Package: DMAR (via r-universe)

October 15, 2026

Type Package

Title Design, Measurement, and Analysis in R (DMAR)

Version 1.0.1

Date 2026-10-15

Imports grDevices, MASS, generics, parallel, stats, utils, withr

Suggests boot, car, ggplot2 (>= 3.4.0), ggrain, knitr, lavaan (>= 0.7-2), lme4, lmerTest, mvtnorm, nlme, OpenMx, patchwork, reformulas, rmarkdown, testthat (>= 3.2.0)

Description Methods for design, measurement, and analysis, with the aim of being user friendly yet methodologically sound. 'DMAR' (pronounced ``Dee-Mar'') implements many advanced and nonstandard methods and makes them available for straightforward use, with interfaces, defaults, and documentation that are consistent across the package and grounded in the methodological literature, in support of sound and reproducible results. The package emphasizes effect size estimation with confidence intervals; sample size planning through accuracy in parameter estimation (AIPE) and power analysis (including composite power for designs whose conclusions require several results to hold at once), with minimum risk, sequential, and equivalence frameworks; reliability, agreement, and measurement more broadly, from coefficient omega with confidence intervals to measurement invariance; factor analysis and structural equation modeling, in which constructs, latent variables measured by multiple indicators, are modeled directly, with confirmatory factor analysis, convergent and discriminant validity, and sample size planning for structural equation models; mediation analysis, from the simple mediation model with bootstrap intervals to likelihood ratio tests of arbitrary indirect effects by model-based constrained optimization (MBCO), with multiple groups and the probing of moderated mediation; equivalence and noninferiority testing; meta-analysis; repeated measures, multivariate, ANOVA, and ANCOVA designs; and inference grounded in model comparison throughout. Measurement

is approached from a psychometric perspective, and although many of the methods grew up in human-centered research, they apply broadly across the empirical sciences. Much of what is implemented traces to the author's methodological work, interests, and collaborations. 'DMAR' is a more modern, more general, and greatly expanded reimagining of the 'MBESS' package (Kelley, 2007a, <doi:10.18637/jss.v020.i08>; 2007b, <doi:10.3758/BF03192993>), which has been on CRAN for more than two decades and remains available there in stable form. Most functions accept either raw data or the summary statistics typically reported in published articles, so an analysis can be reproduced from a paper without the original data, which is useful both for extending a published analysis and for meta-analytic work. The estimation, inference, and planning functions return one consistently formatted data frame per function that composes with the broader R ecosystem, and confidence intervals are reported alongside effect sizes throughout, as best practice recommends. Researchers who have data and a question but who are not R experts will find the package approachable, while methodologists gain access to advanced and nonstandard methods, including tables of critical values not available elsewhere.

Depends R (>= 4.0.0)

URL https://kenkelley.org, https://yelleknek.github.io/DMAR/, https://github.com/yelleKneK/DMAR

BugReports https://github.com/yelleKneK/DMAR/issues

License GPL (>= 3)

Encoding UTF-8

Language en-US

LazyData true

LazyDataCompression xz

VignetteBuilder knitr

Config/testthat/edition 3

Config/testthat/parallel true

Config/testthat/start-first ss_aipe_crd_es, reliability_omega_categorical, cv_bryant_paulson, ci_dunnett, reliability_omega, ss_power_composite_sem, ss_aipe_composite_sem, mediation_mbco, ss_aipe_new_sensitivities

Config/roxygen2/version 8.1.0

Repository https://yelleknek.r-universe.dev

Date/Publication 2026-10-15 12:05:30 UTC

RemoteUrl https://github.com/yelleknek/dmar

RemoteRef HEAD

RemoteSha 4e3b797247bd4f0ff4a25b1116d8fe4a2b0d0754

Contents

DMAR-package . . . 10
adjusted_means . . . 12
analysis_of_change . . . 15
ancova . . . 18
anova.mlmr . . . 20
anova.mlmr_mv . . . 22
anova_within . . . 23
anova_within_two_way . . . 24
average_variance_extracted . . . 27
bayes_independent_t . . . 29
bayes_one_sample_t . . . 32
bayes_paired_t . . . 34
bessel_errors . . . 36
bifactor_indices . . . 39
bryant_paulson . . . 41
cfa_1 . . . 44
cfa_2 . . . 47
cfa_k . . . 50
ci_c . . . 56
ci_c_ancova . . . 59
ci_c_ancova_bp . . . 62
ci_correlation . . . 64
ci_cv . . . 68
ci_dunnett . . . 70
ci_eigenvalue . . . 72
ci_eta_squared . . . 73
ci_eta_squared_generalized . . . 77
ci_eta_squared_partial . . . 80
ci_games_howell . . . 83
ci_mahalanobis . . . 85
ci_nc_chisq . . . 88
ci_nc_F . . . 90
ci_nc_t . . . 92
ci_omega_squared . . . 95
ci_proportion . . . 97
ci_pvaf . . . 99
ci_R2 . . . 101
ci_rc . . . 104
ci_reg_coef . . . 106
ci_rmsea . . . 109
ci_sc . . . 111

ci_sc_ancova . . . 113
ci_scheffe . . . 116
ci_sm . . . 118
ci_smd . . . 120
ci_smd_c . . . 123
ci_snr . . . 125
ci_src . . . 127
ci_srsnr . . . 129
ci_tukey_kramer . . . 131
cles . . . 133
cliff_delta . . . 135
cohen_f . . . 137
cohen_h . . . 139
cohen_kappa . . . 140
combine_p . . . 145
common_method_marker . . . 147
common_method_single_factor . . . 149
compare_cov_structures . . . 151
content_validity_index . . . 153
contrast_adjusted . . . 156
contrast_test . . . 158
convert_cor_cov . . . 161
convert_d_or . . . 163
convert_d_r . . . 164
convert_F_chisq . . . 165
convert_r_Z . . . 168
convert_R2 . . . 169
convert_t_smd . . . 170
convert_z_normal . . . 172
convert_Z_r . . . 173
correction_for_attenuation . . . 174
correlations_test . . . 177
cov_sem . . . 180
covmat_from_cfa . . . 182
cv . . . 184
cv_bonferroni_f . . . 186
cv_bryant_paulson . . . 188
cv_chisq . . . 191
cv_dunnett . . . 192
cv_f . . . 194
cv_scheffe . . . 196
cv_smm . . . 198
cv_t . . . 200
cv_tukey_hsd . . . 201
cv_z . . . 203
depression_bdi . . . 204
descriptives . . . 205
design_consequences . . . 207

design_effect . . . 211
diagnosis_agreement . . . 213
dmacs . . . 215
dmar_output_helpers . . . 218
drinks_trial . . . 220
dunn_test . . . 223
ecvi . . . 226
effects_coding . . . 227
epsilon_corrections . . . 229
equivalence_c . . . 230
equivalence_r . . . 234
equivalence_smd . . . 236
eta_squared . . . 238
eta_squared_generalized . . . 241
eta_squared_partial . . . 245
expected_partial_r . . . 247
expected_r . . . 249
expected_R2 . . . 253
expected_smd . . . 254
factorial_anova . . . 256
fleiss_kappa . . . 259
format_p . . . 262
glance.dmar_cfa_k . . . 263
glance.dmar_mediation_mbco . . . 264
glance.dmar_reliability . . . 265
glance.mlmr . . . 265
gwet_ac . . . 266
helmert_coding . . . 268
holzinger_swineford . . . 269
htmt . . . 272
icc . . . 274
icc_lmer . . . 276
irt_grm . . . 278
irt_information . . . 282
is_orthogonal_set . . . 286
krippendorff_alpha . . . 287
kurtosis . . . 290
limits_of_agreement . . . 291
lin_ccc . . . 294
manova_split_plot . . . 296
mauchly_test . . . 298
measurement_alignment . . . 300
measurement_invariance . . . 305
mediate . . . 312
mediation_mbco . . . 315
meta_contrast . . . 323
meta_es . . . 325
meta_r . . . 327

meta_smd . . . 329
mixed_anova . . . 331
mlmr . . . 333
mlmr_mv . . . 339
moments_nc_chisq . . . 344
moments_nc_F . . . 345
moments_nc_t . . . 347
mr_cv . . . 349
mr_smd . . . 351
nnt_from_smd . . . 354
now . . . 356
obrien_test . . . 358
omega_squared . . . 360
omega_squared_partial . . . 363
orthogonal_polynomial . . . 366
pairwise_within . . . 369
plot_cfa_k . . . 371
plot_ci . . . 372
plot_equivalence . . . 375
plot_forest . . . 377
plot_irt_information . . . 379
plot_mediation_mbco . . . 381
plot_R2 . . . 384
plot_randomization_test . . . 386
plot_regions_of_significance . . . 387
plot_smd . . . 389
plot_trajectories . . . 392
plot_trajectories_fitted . . . 394
power_density_equivalence_md . . . 397
power_equivalence_c . . . 398
power_equivalence_md . . . 401
power_equivalence_md_plot . . . 403
power_fisher_exact . . . 405
prime_time_achievement . . . 407
print_anova . . . 416
print_summary . . . 417
probability_of_superiority_paired . . . 418
procrustes_phi . . . 420
proportion_of_superiority . . . 422
pygmalion . . . 423
R2_mixed_effects . . . 426
R2_mixed_effects_decomposition . . . 429
randomization_test . . . 431
randomization_test_paired . . . 435
regions_of_significance . . . 438
reliability . . . 441
reliability_alpha . . . 445
reliability_H . . . 452

reliability_kr20 . . . 454
reliability_omega . . . 457
reliability_omega_categorical . . . 465
responder_analysis . . . 469
sd_unbiased . . . 471
signal_to_noise_R2 . . . 473
simple_effects_AB . . . 475
simple_structure . . . 478
simulate_ancova_data . . . 480
simulate_ancova_factorial_data . . . 483
simulate_anova_data . . . 486
simulate_longitudinal_gompertz . . . 487
simulate_longitudinal_logistic . . . 491
simulate_longitudinal_negative_exponential . . . 495
simulate_longitudinal_polynomial . . . 498
simulate_longitudinal_richards . . . 504
simulate_regression_data . . . 508
skewness . . . 510
smd . . . 512
smd_c . . . 514
smd_trimmed . . . 516
ss_aipe_c . . . 518
ss_aipe_c_ancova . . . 520
ss_aipe_c_ancova_sensitivity . . . 522
ss_aipe_c_sensitivity . . . 524
ss_aipe_cliff_delta . . . 526
ss_aipe_cliff_delta_sensitivity . . . 528
ss_aipe_composite_sem . . . 530
ss_aipe_crd . . . 535
ss_aipe_crd_es . . . 540
ss_aipe_cv . . . 547
ss_aipe_cv_sensitivity . . . 548
ss_aipe_equivalence_r . . . 551
ss_aipe_equivalence_r_sensitivity . . . 553
ss_aipe_equivalence_smd . . . 556
ss_aipe_equivalence_smd_sensitivity . . . 558
ss_aipe_icc . . . 560
ss_aipe_icc_sensitivity . . . 563
ss_aipe_indirect_effect . . . 566
ss_aipe_indirect_effect_sensitivity . . . 571
ss_aipe_mixed_effects . . . 573
ss_aipe_mixed_effects_sensitivity . . . 575
ss_aipe_omega_squared . . . 577
ss_aipe_omega_squared_sensitivity . . . 580
ss_aipe_partial_r . . . 582
ss_aipe_partial_r_sensitivity . . . 585
ss_aipe_pcm . . . 587
ss_aipe_pcm_sensitivity . . . 590

ss_aipe_r . . . 592
ss_aipe_r_sensitivity . . . 594
ss_aipe_R2 . . . 596
ss_aipe_R2_sensitivity . . . 599
ss_aipe_rc . . . 602
ss_aipe_rc_sensitivity . . . 604
ss_aipe_reg_coef . . . 607
ss_aipe_reg_coef_sensitivity . . . 611
ss_aipe_reliability . . . 614
ss_aipe_reliability_sensitivity . . . 618
ss_aipe_rmsea . . . 620
ss_aipe_rmsea_sensitivity . . . 621
ss_aipe_sc . . . 624
ss_aipe_sc_ancova . . . 626
ss_aipe_sc_ancova_sensitivity . . . 628
ss_aipe_sc_sensitivity . . . 631
ss_aipe_sem_path . . . 633
ss_aipe_sem_path_sensitivity . . . 636
ss_aipe_semipartial_r . . . 639
ss_aipe_semipartial_r_sensitivity . . . 641
ss_aipe_sm . . . 643
ss_aipe_sm_sensitivity . . . 645
ss_aipe_smd . . . 647
ss_aipe_smd_sensitivity . . . 649
ss_aipe_src . . . 652
ss_aipe_src_sensitivity . . . 655
ss_power_c . . . 657
ss_power_c_ancova . . . 659
ss_power_composite_ancova . . . 661
ss_power_composite_ancova_2group . . . 665
ss_power_composite_anova . . . 672
ss_power_composite_factorial_ancova . . . 675
ss_power_composite_factorial_ancova_het . . . 679
ss_power_composite_factorial_anova . . . 683
ss_power_composite_sem . . . 686
ss_power_contrast . . . 691
ss_power_equivalence_c . . . 694
ss_power_factorial_ancova . . . 696
ss_power_factorial_anova . . . 699
ss_power_indirect_effect . . . 701
ss_power_mixed_effects . . . 703
ss_power_one_way_anova . . . 705
ss_power_pcm . . . 707
ss_power_r . . . 711
ss_power_R2 . . . 713
ss_power_R2_sensitivity . . . 716
ss_power_rc . . . 719
ss_power_reg_coef . . . 723

ss_power_reg_coef_sensitivity . . . 727
ss_power_rm_anova . . . 730
ss_power_sc . . . 732
ss_power_sem . . . 734
ss_power_smd . . . 736
ss_power_split_plot_anova . . . 739
ss_seq_c . . . 741
ss_seq_c_sensitivity . . . 744
summary_t_test . . . 746
summary_t_test_broom . . . 748
teacher_expectancy . . . 749
test_market . . . 751
tidy.dmar_cfa_k . . . 753
tidy.dmar_ci_R2 . . . 754
tidy.dmar_ci_smd . . . 755
tidy.dmar_contrast_test . . . 756
tidy.dmar_mediation_mbco . . . 760
tidy.dmar_R2_mixed_effects . . . 761
tidy.dmar_reliability . . . 762
tidy.mlmr . . . 763
tidy.mlmr_mv . . . 764
unbiased_R2 . . . 765
var_alpha . . . 767
var_cv . . . 769
var_ete . . . 770
var_icc . . . 773
var_indirect_effect . . . 775
var_omega_squared . . . 777
var_partial_r . . . 780
var_r . . . 782
var_R2 . . . 784
var_semipartial_r . . . 785
var_smd . . . 787
var_smd_trimmed . . . 789
vargha_delaney_A . . . 791
variance_components_mls . . . 794
welch_t . . . 796
welch_t_broom . . . 798

Index **800**

DMAR-package *Design, Measurement, and Analysis in R*

Description

A modern R package for design, measurement, and analysis, with special strength in effect sizes, confidence intervals, size planning, reliability and agreement, mediation analysis, equivalence testing, meta-analysis, experimental and quasi-experimental designs, repeated measures, and model comparison-based inference. DMAR (pronounced "Dee-Mar") is heavily methodological in nature, drawing on the psychometric and statistical traditions, and is aligned with the methodological and applied research program and interests of the author. Many aspects of the package traces to the author's methodological work and collaborations, including sample size planning via accuracy in parameter estimation (AIPE; Kelley & Maxwell, 2003; Kelley & Rausch, 2006; Maxwell, Kelley, & Rausch, 2008), the definition and communication of effect sizes (Kelley & Preacher, 2012; Preacher & Kelley, 2011), and the model comparison perspective of Maxwell, Delaney, and Kelley (2027). It aims to be methodologically sound and particularly well-suited to research in which the independent or dependent variables involve the person, across psychology, sociology, education, behavioral economics, management, marketing, and information systems.

Details

The package makes accessible to researchers a variety of methods that are easy to use, including from sample estimates or results reported in published articles: effect size estimation, confidence intervals for effect sizes, sample size planning, multivariate methods, factor analysis, and certain latent variable models. Particular strengths include sample size planning under several complementary frameworks: accuracy in parameter estimation (AIPE), power analysis, minimum risk, and equivalence. Most exported functions return a tidy `data.frame` with a `term` column and a numeric `value` column (some carry additional typed columns per term), and **ggplot2** is used for the plotting functions. A few functions return the shape their task calls for instead: model fits such as `mlmr` return a richer list-like object with `coef` / `vcov` / `confint` methods, `descriptives` returns a list of summary tables, and a small number of scalar utilities such as `skewness` and `kurtosis` return a bare numeric. The interface is consistent, modern, and opinionated, and is designed for clarity and reproducibility.

DMAR builds heavily on the **MBESS** package (Kelley, 2007a, 2007b), which has been on CRAN for more than two decades. **MBESS** was originally framed for the behavioral, educational, and social sciences, but its use has grown well beyond that scope; DMAR is quite general, though especially well-aligned with human-centered research.

Function families. A user-facing tour:

Effect sizes `smd`, `smd_c`, `smd_trimmed`, `eta_squared`, `eta_squared_partial`, `eta_squared_generalized`, `omega_squared`, `omega_squared_partial`, `cohen_f`, `cles`, `cliff_delta`, `vargha_delaney_A`, `proportion_of_superiority`, `probability_of_superiority_paired`, `lin_ccc`.

Confidence intervals on effect sizes `ci_smd`, `ci_smd_c`, `ci_R2`, `ci_R`, `ci_rc`, `ci_src`, `ci_eta_squared` (and partial / generalized variants), `ci_omega_squared`, `ci_pvaf`, `ci_snr`, `ci_srsnr`, `ci_mahalanobis`, `ci_eigenvalue`, `ci_cv`, `ci_sm`, `ci_reg_coef`, `ci_r`, `ci_rmsea`.

Maximum likelihood regression `mlmr` (univariate full information maximum likelihood (FIML), lm-like), `mlmr_mv` (multivariate FIML).

ANOVA and ANCOVA `ancova`, `anova_within_two_way`, `mixed_anova`, `manova_split_plot`, `simple_effects_AB`, `contrast_test`, `pairwise_within`, `mauchly_test`, `obrien_test`.

Reliability and agreement `reliability`, `reliability_alpha`, `reliability_omega` (with a model implied or observed total-variance denominator, and a `reliability_omega_categorical` for ordered items), `reliability_kr20`, `reliability_H`, `cohen_kappa`, `fleiss_kappa`, `krippendorff_alpha`, `gwet_ac`, `limits_of_agreement`.

Mediation `mediate` (the simple mediation model with bootstrap, Monte Carlo, and Sobel intervals), `mediation_mbco` (likelihood ratio tests of arbitrary mediation effects by model-based constrained optimization, with multiple groups and moderated mediation probing), and `plot_mediation_mbco` (conditional effect curves with confidence bands).

Confirmatory factor and SEM tools `cfa_1`, `cov_sem`, `covmat_from_cfa`, `compare_cov_structures`.

Sample size planning (AIPE) `ss_aipe_smd`, `ss_aipe_R2`, `ss_aipe_reg_coef`, `ss_aipe_partial_r`, `ss_aipe_omega_squared`, `ss_aipe_icc`, `ss_aipe_cv`, `ss_aipe_pcm`, `ss_aipe_rmsea`, the cluster-randomized planners `ss_aipe_crd_*`, plus their Monte Carlo sensitivity companions `ss_aipe_*_sensitivity`.

Sample size planning (power) `ss_power_smd`, `ss_power_R2`, `ss_power_r`, `ss_power_reg_coef`, `ss_power_sem`, `ss_power_c`, `ss_power_c_ancova`, `ss_power_contrast`, `ss_power_factorial_anova`, `ss_power_split_plot_anova`, `ss_power_mixed_effects`, `ss_power_one_way_anova`, `ss_power_pcm`, `ss_power_rm_anova`, `ss_power_sc`.

Critical values and tests `cv_t`, `cv_z`, `cv_smm`, `cv_scheffe`, `cv_tukey_hsd`, `cv_dunnett`, `ci_dunnett`, `ci_tukey_kramer`, `ci_scheffe`, `welch_t`, `summary_t_test`, `correlations_test`, `power_fisher_exact`, `randomization_test_paired`, `equivalence_smd`, `equivalence_r`, `power_equivalence_md`.

Parameterization conversions `convert_R2_f` / `convert_f_R2`, `convert_R2_lambda` / `convert_lambda_R2`, `convert_delta_lambda` / `convert_lambda_delta`, `convert_r_Z` / `convert_Z_r`, `convert_cor_cov`.

Visualization `plot_smd`, `plot_ci`, `plot_R2`, `plot_trajectories`, `plot_trajectories_fitted`.

Multilevel and clustering `icc`, `icc_lmer`, `variance_components_mls`, `design_effect` (Kish design effect), `ss_aipe_crd_*`.

Data sets `bessel_errors` (Bessel's 1818 grouped distribution of Bradley's astronomical observation errors), `diagnosis_agreement` (Cohen's 1968 weighted kappa illustration), `drinks_trial` (Smith, Meyers, and Delaney's 1998 Community Reinforcement Approach drinking trial), `holzinger_swineford` (the 1939 factor analysis study), `prime_time_achievement` (the Indiana Prime Time third grade achievement evaluation), `pygmalion` (Rosenthal and Jacobson's 1968 teacher-expectancy data), `teacher_expectancy` (Raudenbush's 1984 meta-analysis of 18 teacher-expectancy experiments), and `test_market` (Bryant and Bruvold's 1980 controlled test-market experiment for ANCOVA with a random covariate).

Feedback. Bug reports, feature requests, and suggestions for new methods are welcomed by email to Ken Kelley <kkelley@nd.edu> (please put "DMAR" in the subject line). See https://kenkelley.org for Ken Kelley's web site, https://kenkelley.org/publications/ for related publications, and https://github.com/yelleKneK/DMAR for the project's GitHub page.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007a). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K. (2007b). Methods for the behavioral, educational, and social sciences: An R package. *Behavior Research Methods, 39*(4), 979–984. doi:10.3758/BF03192993

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Kelley, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*(2), 137–152. doi:10.1037/a0028086

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

Preacher, K. J., & Kelley, K. (2011). Effect size measures for mediation models: Quantitative strategies for communicating indirect effects. *Psychological Methods, 16*(2), 93–115. doi:10.1037/a0022658

See Also

Useful links:

- https://kenkelley.org
- https://yelleknek.github.io/DMAR/
- https://github.com/yelleKneK/DMAR
- Report bugs at https://github.com/yelleKneK/DMAR/issues

`adjusted_means` *Adjusted Cell and Marginal Means From a Fitted Linear Model*

Description

Given a fitted `lm` or `aov` object with one or more factors among its predictors, `adjusted_means()` returns the means the model actually compares, sometimes called least-squares means or estimated marginal means. By default the table has one row per cell of the crossed factor design, each cell's mean being the model's predicted response at that combination of factor levels with every covariate held at its sample mean (the adjusted cell means of an ANCOVA; for a model without covariates, the model-based cell means). Naming one or more factors in by instead returns the marginal means of those factors, formed by averaging the cell predictions over the remaining factors with either equal or frequency-proportional weights. Every mean is accompanied by its standard error and a *t* confidence interval on the model's residual degrees of freedom.

Usage

```
adjusted_means(
  model,
  by = NULL,
  weights = c("equal", "proportional"),
  conf_level = 0.95
)
```

Arguments

model	A fitted lm or aov object with one or more factors (and optionally covariates) on the right-hand side of the formula.
by	NULL (default) for the cell means table, or a character vector naming one or more of the model's factors for their marginal means. The output rows cross the named factors in the order given, first factor varying fastest.
weights	Weighting used to average cell predictions into marginal means, so it matters only when by is supplied and the data are unbalanced. "equal" (default) weights every combination of the averaged-over factors equally; "proportional" weights each combination by its observed frequency.
conf_level	The confidence level for the intervals (default 0.95).

Details

The reference grid and adjusted cell means. The reference grid is the crossing of the model's factor levels, enumerated in the order the factors appear in the model formula with the first factor varying fastest (the order expand.grid produces). This is the same cell order contrast_adjusted expects, so contrast weights can be read off this table row by row. Every covariate enters the grid at its sample mean, and a transformed covariate is evaluated by applying the transformation to the mean of the raw variable: with log(x) in the formula the grid carries mean(x) and the model matrix applies log(), and a poly(x, 2) basis is evaluated at $\bar{x}$, matching predict on new data at the covariate mean. Writing L for the matrix whose rows are the design-matrix rows of the grid cells, the cell means are $L\hat{\beta}$, each standard error is the square root of the corresponding diagonal element of $L\,\mathrm{vcov}(\hat{\beta})\,L'$, and each interval is the t interval on the model's residual degrees of freedom.

Marginal means and the two weightings. With by, the cell predictions are averaged over the factors not named there, and the averaging happens in the coefficient map itself: the marginal mean's L row is the weighted average of its cells' rows, so the estimate and the standard error both follow from one linear function of the coefficients. weights = "equal" weights every combination of the averaged-over factors equally; this is the population marginal mean of Searle, Speed, and Milliken (1980), the mean for a population in which every cell is equally represented regardless of the sample's cell sizes. weights = "proportional" weights each averaged-over combination by its observed frequency (in a weighted fit, by its total prior weight), so the marginal mean targets a population whose margins are shaped like the sample's. With balanced data the two weightings coincide; with unbalanced data they generally differ, and the choice between them is a substantive question about the population of interest, not a technical one (Maxwell, Delaney, and Kelley, 2027, Chapter 7).

Nonestimable means. When the fitted design is rank deficient (for example an empty factorial cell), the model has no predicted value for the affected cell, and a marginal mean that averages over

such a cell does not exist either. adjusted_means() refuses with an error naming the affected rows rather than reporting a value contaminated by lm's arbitrary zero for the aliased coefficient.

Scope. The function covers single-stratum lm and aov fits with a single response. Multi-stratum aovlist fits (within-subjects designs fit with an Error() term) are refused, because a within-subjects marginal mean takes its standard error from the matching error stratum, which this function does not compute. Factors must enter the model as variables in the data, not as conversions inside the formula: y ~ factor(g) + x is refused, so convert g in the data first.

Value

A data.frame (class dmar_tbl) with one row per cell of the reference grid or, with by, one row per combination of the named factors. The leading columns give the factor levels; the numeric columns are estimate (the adjusted mean), se (its standard error), and ci_lower / ci_upper (the *t* confidence limits). The residual degrees of freedom of the intervals are attached as the df_residual attribute and, when by is supplied, the weighting as the weights attribute. The stored values keep full precision; only the display rounds (see dmar_tbl).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 7 on nonorthogonal factorial designs and Chapter 9 on designs with covariates.)

Searle, S. R., Speed, F. M., & Milliken, G. A. (1980). Population marginal means in the linear model: An alternative to least squares means. *The American Statistician, 34*(4), 216–221.

See Also

contrast_adjusted for a confidence interval on a single contrast of the adjusted cell means; ancova for the one-way ANCOVA table; ci_dunnett for simultaneous many-to-one comparisons.

Other hypothesis tests: ancova(), anova_within(), ci_dunnett(), ci_scheffe(), ci_tukey_kramer(), compare_cov_structures(), contrast_test(), correlations_test(), equivalence_r(), equivalence_smd(), factorial_anova(), manova_split_plot(), mauchly_test(), mixed_anova(), obrien_test(), pairwise_within(), randomization_test(), randomization_test_paired(), regions_of_significance(), simple_effects_AB(), summary_t_test(), welch_t()

Examples

```
# 1. Cell means of a 2 x 3 factorial (no covariate): end-of-study IQ in
#    the pygmalion expectancy experiment, grades 1 through 3. The grade
#    factor is created in the data, not inside the formula.
pyg <- subset(pygmalion, grade <= 3)
pyg$grade <- factor(pyg$grade)
fit <- lm(iq_8 ~ treatment * grade, data = pyg)
adjusted_means(fit)
```

```
# 2. Marginal means of grade, averaging the cell means over treatment.
adjusted_means(fit, by = "grade")

# 3. An ANCOVA: each adjusted mean holds the covariate, here the pretest
#    depression score, at its sample mean.
fit_ancova <- lm(bdi_post ~ condition + bdi_pre, data = depression_bdi)
adjusted_means(fit_ancova)

# 4. With unbalanced cells (few bloomers in every grade) the two
#    weightings answer different questions.
adjusted_means(fit, by = "treatment")
adjusted_means(fit, by = "treatment", weights = "proportional")
```

`analysis_of_change` *Analysis of Change: Fit Change Models to One or Many Trajectories*

Description

Fits a change model to longitudinal data: any of the package's four nonlinear change models (negative exponential, logistic, Gompertz, Richards; the parameterizations of Kelley, 2005, 2008, exactly as generated by the `simulate_longitudinal_*()` simulators) or a polynomial of any order (the linear change model of `simulate_longitudinal_polynomial`). Two estimation methods are offered. The default, `method = "two_stage"`, fits each unit's curve separately, using only that unit's data, and then summarizes the unit-level parameters: their mean, their standard deviation and variance across units (the individual differences), and the standard error of the mean. `method = "mixed"` fits the proper random-coefficients mixed-effects model simultaneously, with every parameter carrying a random effect, so the reported between-unit spread is a variance component purged of estimation noise. With a single trajectory ($N = 1$, or `id = NULL`) the two-stage method reduces to one least squares fit of that unit's change, reported with its standard errors.

Usage

```
analysis_of_change(
  data,
  id,
  time,
  outcome,
 model = c("negative_exponential", "logistic", "gompertz", "richards", "polynomial"),
  method = c("two_stage", "mixed"),
  order = 1L,
  start = NULL,
  maxiter = 500L
)
```

Arguments

`data`	A `data.frame` in long format: one row per observation, as returned by the `simulate_longitudinal_*()` simulators.

id	Name of the column identifying units (persons, animals, trees, classrooms). NULL treats all rows as a single trajectory (method = "two_stage" only).
time	Name of the time column.
outcome	Name of the outcome column.
model	Which change model to fit: "negative_exponential" (parameters alpha, zeta, gamma), "logistic" (alpha, beta, gamma, zeta), "gompertz" (alpha, beta, gamma, zeta), "richards" (alpha, beta, gamma, delta, zeta), or "polynomial" (coefficients b0 through bP, with the order P set by order). See the corresponding simulator's help page for what each parameter means.
method	How the model is estimated. "two_stage" (the default) fits one curve per unit, each from that unit's data alone, and summarizes across units; it is transparent, works at $N = 1$, and never lets one unit's data influence another's fit. "mixed" estimates the random-coefficients model simultaneously: lme4::lmer() for the polynomial (a linear mixed model) and nlme::nlme() for the nonlinear curves (started at the two-stage estimates), with uncorrelated random effects on every parameter. See Details for how to choose.
order	Polynomial order P when model = "polynomial" (1 is straight-line change, 2 quadratic, and so on). Default 1. Ignored for the nonlinear models.
start	Optional named numeric vector of starting values for a nonlinear model's parameters, used for every unit (and, under method = "mixed", for the fixed effects). NULL (default) derives data-driven starting values per unit (see Details). Ignored for the polynomial, whose fit is closed form.
maxiter	Maximum number of iterations passed to nls (and, under method = "mixed", to the nonlinear mixed-effects optimizer). Default 500.

Details

Choosing between the methods. The two-stage (curve-by-curve) route is the transparent classic: every unit's curve is inspectable, no unit's data influence another's fit, and it is the only method available for a single trajectory. Its known cost is that sd_units reflects the spread of *estimates*, which adds each fit's estimation noise to the true individual differences; with short or noisy trajectories it therefore overstates the population standard deviation. The mixed method estimates that between-unit variation as a variance component, separating it from level-one error, and borrows strength across units, at the price of a harder estimation problem (and, for the nonlinear curves, occasional convergence trouble; the fit is started at the two-stage estimates, and a failure suggests simplifying the model or falling back to two-stage). Under method = "mixed" the random effects are uncorrelated across parameters, matching the simulators' default.

Relation to existing tools. The two-stage method is the idea behind nlme::lmList() and nlme::nlsList(), and the mixed method wraps lme4::lmer() and nlme::nlme(); base R also ships self-starting curves (SSasymp, SSfpl, SSgompertz) in other parameterizations. What this function adds is the package's landmark parameterizations (the intercept-shifting ζ floor, the Richards δ; SSgompertz's $a\exp(-b_2 b_3^x)$ answers no substantive question directly), the exact match to the simulate_longitudinal_*() simulators so design studies close the loop, one interface across linear and nonlinear change, and the package's tidy summary with failed fits dropped under a single counted warning.

Units whose two-stage fit does not converge are dropped with a single warning reporting how many, and the effective count is the "n_used" attribute. If no unit's fit converges the function stops. The polynomial fit is closed form and does not fail on any trajectory with at least $P + 1$ occasions.

Starting values. Unless `start` is supplied, each unit's nonlinear starting values are derived from that unit's data: the floor and the span from early and late observations, the inflection time from where the trajectory crosses the middle of its range, the curvature from the time the trajectory needs to travel the central half of its range, and, for the Richards model, a logistic start (delta = 1). Cleanly measured trajectories rarely need more; hard cases (very short series, strong decreasing curves, near-flat change) may need an explicit `start`.

Value

A `data.frame` with one row per model parameter and columns

`term` The parameter name.

`estimate` The mean of the unit-level estimates (two-stage; with one trajectory, that unit's estimate) or the fixed effect (mixed).

`se` The standard error of `estimate`: the between-unit standard deviation over the square root of the number of fitted units (two-stage), the asymptotic standard error of the single fit (one trajectory), or the fixed-effect standard error (mixed).

`sd_units` The between-unit standard deviation of the parameter: the spread of the unit-level estimates (two-stage; `NA` for a single trajectory) or the random-effect standard deviation (mixed).

`var_units` The corresponding variance.

Attributes: `"model"`, `"method"`, `"n_units"` (units supplied), `"n_used"` (units whose fit converged; equal to `"n_units"` under `method = "mixed"`), `"sigma"` (the level-one residual standard deviation), and `"per_unit_estimates"` (a units-by-parameters matrix: the separate unit-level estimates under two-stage, or the unit-level predictions `coef()` under mixed, which are shrunken toward the fixed effects).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2005). *Estimating nonlinear change models in heterogeneous populations when class membership is unknown: Defining and developing the latent classification differential change model* (Doctoral dissertation). University of Notre Dame.

Kelley, K. (2008). Nonlinear change models in populations with unobserved heterogeneity. *Methodology, 4*(3), 97–112.

Pinheiro, J. C., & Bates, D. M. (2000). *Mixed-effects models in S and S-PLUS*. Springer.

Richards, F. J. (1959). A flexible growth function for empirical use. *Journal of Experimental Botany, 10*(2), 290–301.

See Also

The generators `simulate_longitudinal_negative_exponential`, `simulate_longitudinal_logistic`, `simulate_longitudinal_gompertz`, `simulate_longitudinal_richards`, `simulate_longitudinal_polynomial`; `plot_trajectories` for plotting the data being fit; `nlme::lmList()`, `nlme::nlsList()`, `lme4::lmer()`, and `nlme::nlme()` for the engines and their generalizations.

Examples

```
# Simulate a Gompertz population with individual differences, then
# recover both the mean curve and the spread of its parameters.
set.seed(113)
d <- simulate_longitudinal_gompertz(
  n = 40, target_times = 0:10,
  fixed_parameters = c(alpha = 75, beta = 3, gamma = 0.55, zeta = 10),
  random_variances = c(alpha = 25, beta = 0.4, gamma = 0.005, zeta = 4),
  error_variance = 4
)
analysis_of_change(d, id = "id", time = "time", outcome = "y",
                   model = "gompertz")

# The same data as a straight-line (order 1) polynomial: the linear
# model has nothing to say about floors, ceilings, or timing.
analysis_of_change(d, id = "id", time = "time", outcome = "y",
                   model = "polynomial", order = 1)

# A single unit's trajectory: N = 1 is one fit, with asymptotic
# standard errors in place of between-unit spread.
one <- d[d$id == levels(d$id)[1], ]
analysis_of_change(one, id = NULL, time = "time", outcome = "y",
                   model = "gompertz")

# The proper mixed-effects model, fit simultaneously across units:
# nlme::nlme() for a nonlinear curve, lme4::lmer() for the polynomial
# (the latter requires lme4 to be installed).
analysis_of_change(d, id = "id", time = "time", outcome = "y",
                   model = "gompertz", method = "mixed")
analysis_of_change(d, id = "id", time = "time", outcome = "y",
                   model = "polynomial", order = 1,
                   method = "mixed")
```

---

ancova *Analysis of Covariance (ANCOVA)*

Description

Fits a one-way analysis of covariance so the covariate-adjusted group comparison is available from a single call, without assembling the adjusted means, the omnibus test, and the effect sizes by hand. It returns the adjusted (covariate-corrected) group means with standard errors, the covariate-adjusted omnibus F for the group effect (Type III sums of squares), partial η^2 and partial ω^2 with noncentral F confidence intervals, and a homogeneity-of-regression check, returned in one `data.frame`.

Usage

```
ancova(data, outcome, treatment, covariates, conf_level = 0.95)
```

Arguments

`data`	A `data.frame` containing the response, the treatment factor, and the covariate(s).
`outcome`	Character name of the response column.
`treatment`	Character name of the grouping factor column (the groups being compared, for example treatment arms); a factor or character column.
`covariates`	Character vector of one or more covariate column names.
`conf_level`	Confidence level for the effect size CIs. Default `0.95`.

Details

Covariate-adjusted test (Type III sums of squares). The omnibus F tests the group effect after adjusting for the covariate(s), that is, the Type III sum of squares for the grouping factor. For a one-way ANCOVA (one grouping factor, with the covariate slopes held constant) the Type II and Type III sums of squares for the group effect coincide, and both equal the sequential sum of squares obtained with the covariate(s) entered first and the grouping factor last, which is how it is computed here; the value matches `car::Anova(fit, type = 3)`. The choice of sum-of-squares type changes the result only in designs with more than one factor or with interactions among factors (Maxwell, Delaney, and Kelley, 2027, Chapter 7); for those, the two-way and mixed analyses report their sum-of-squares type and allow Type I, II, or III.

Adjusted means. The adjusted mean for treatment level j is the model-predicted response at $X = \bar{X}$ (the covariate grand mean):

$$\hat{\mu}_j^{\mathrm{adj}} \;=\; \hat{\mu}_j - \sum_k \hat{\beta}_k (\bar{X}_{kj} - \bar{X}_k),$$

where $\hat{\beta}_k$ is the within-cell slope on covariate k and $\bar{X}_{kj}, \bar{X}_k$ are the per-cell and grand means of covariate k.

Homogeneity of regression. The model fit here holds the within-group covariate slopes β_k constant across groups. This is a property of the particular model being fit, not an assumption of analysis of covariance in general: it is a testable claim. Adding all group-by-covariate interactions gives an expanded model, and a model comparison F-test of the additive model against the expanded one (`stats::anova`) assesses whether the slopes differ across groups (Maxwell, Delaney, and Kelley, 2027, Chapter 9). A large F indicates the slopes are not constant, in which case the single adjusted comparison is not the whole story and the interaction model should be entertained directly. The check is reported in the `F_homogeneity_of_regression` rows.

Effect size CIs. Partial η^2 and partial ω^2 use the noncentral F framework (`ci_eta_squared_partial`, `ci_omega_squared`).

Value

A `data.frame` (class `dmar_tbl`) with rows for the omnibus test (`F_value`, `df_1`, `df_2`, `p_value`), the sum-of-squares type used (`sum_of_squares_type`; 3 for Type III), the point estimates and confidence intervals of partial η^2 and partial ω^2, the adjusted group means and their standard errors (one row per level of `treatment`), and the homogeneity-of-regression F-test. The result carries the `dmar_tbl` class, so it prints to 3 significant figures with whole numbers (such as the degrees of freedom) shown without a decimal part and p-values to 4 decimal places (a p-value below 0.0001

prints as "< 0.0001"); the stored values keep full precision. Control the display with `print(x, digits = )` or globally with `options(dmar.digits = )` (see `dmar_tbl`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Huitema, B. E. (2011). *The analysis of covariance and alternatives* (2nd ed.). Wiley.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9 on analysis of covariance and Chapter 7 on higher-order designs.)

See Also

`ci_c_ancova`, `ci_sc_ancova`, `ss_aipe_sc_ancova`, `omega_squared_partial`

Other hypothesis tests: `adjusted_means()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `correlations_test()`, `equivalence_r()`, `equivalence_smd()`, `factorial_anova()`, `manova_split_plot()`, `mauchly_test()`, `mixed_anova()`, `obrien_test()`, `pairwise_within()`, `randomization_test()`, `randomization_test_paired()`, `regions_of_significance()`, `simple_effects_AB()`, `summary_t_test()`, `welch_t()`

Examples

```
# 1. Compare the two groups in the Pygmalion data on eighth-grade IQ,
#    adjusting for pre-test IQ.
ancova(outcome = "iq_8", treatment = "treatment",
       covariates = "iq_pre", data = pygmalion)

# 2. The same comparison adjusting for two covariates (pre-test IQ and
#    grade).
ancova(outcome = "iq_8", treatment = "treatment",
       covariates = c("iq_pre", "grade"), data = pygmalion)
```

---

anova.mlmr *Likelihood Ratio Test for Nested Mlmr Fits*

Description

Compares two or more nested `mlmr` fits with the likelihood ratio test, delegating the chi square computation to `lavTestLRT`. The models must be nested (every parameter in the more restricted model is also in the more general one) and must be fit to the same data with the same missing data handling.

Usage

```
## S3 method for class 'mlmr'
anova(object, ...)
```

Arguments

object	An mlmr fit.
...	Additional mlmr fits, nested with respect to object.

Details

"The same data" means the same observations, with the variables the fits share holding the same values. It does not mean the same number of complete cases. Under missing = "fiml" a legitimate nested comparison routinely has different complete-case counts: adding a predictor that is itself incompletely observed lowers the number of rows complete on every modeled variable, yet the observations, and the information each of them contributes to the likelihood, are unchanged. Comparing y ~ x1 with y ~ x1 + x2 when x2 has missing values is exactly the comparison full information maximum likelihood exists to support, and it is accepted. Fits run on genuinely different data are refused.

Each smaller model is refit as a constrained version of the largest one on the largest fit's data, with the slopes of its absent predictors fixed at zero. That keeps the comparison on a single joint observed variable set, which is what makes the chi square difference interpretable when the predictors are modeled as random (fixed_x = FALSE, the mlmr default).

Value

A data.frame of class anova reporting the degrees of freedom, AIC, BIC, log-likelihood, chi square test statistic, and *p*-value for each consecutive pairwise comparison.

Author(s)

Ken Kelley <kkelley@nd.edu>

Examples

```
# Both fits ask for the Wald interval and skip the effect size block,
# since the likelihood ratio test needs neither and each costs refits.
fit1 <- mlmr(t6_paragraph_comprehension ~ t5_general_information,
             data = holzinger_swineford, ci_method = "wald",
             effect_sizes = FALSE)
fit2 <- mlmr(t6_paragraph_comprehension ~ t5_general_information +
               t9_word_meaning,
             data = holzinger_swineford, ci_method = "wald",
             effect_sizes = FALSE)
anova(fit1, fit2)
```

anova.mlmr_mv *Compare Nested Multivariate FIML Regression Fits*

Description

Compares two or more nested mlmr_mv fits with the likelihood ratio test, delegating the chi square computation to lavTestLRT. The models must be nested (every predictor in the more restricted model is also in the more general one) and must be fit to the same outcomes and the same data with the same missing data handling. This is the multivariate counterpart of anova.mlmr: because every outcome is regressed on the shared predictor set, dropping a predictor drops its slope on every outcome, so the nesting constraint sets that slope to zero across all outcomes at once.

Usage

```
## S3 method for class 'mlmr_mv'
anova(object, ...)
```

Arguments

object	An mlmr_mv fit.
...	Additional mlmr_mv fits, nested with respect to object.

Details

"The same data" means the same observations, with the variables the fits share holding the same values. It does not mean the same number of complete cases. Under missing = "fiml" a legitimate nested comparison routinely has different complete-case counts: adding a predictor that is itself incompletely observed lowers the number of rows complete on every modeled variable, yet the observations, and the information each of them contributes to the likelihood, are unchanged. That comparison is accepted. Fits run on genuinely different data are refused.

Value

A data.frame of class anova reporting the degrees of freedom, AIC, BIC, log-likelihood, chi square test statistic, and *p*-value for each consecutive pairwise comparison.

Author(s)

Ken Kelley <kkelley@nd.edu>

Examples

```
# Both fits ask for the Wald interval and skip the effect size block,
# since the likelihood ratio test needs neither and each costs refits.
fit1 <- mlmr_mv(cbind(t6_paragraph_comprehension, t9_word_meaning) ~
                  t5_general_information,
                data = holzinger_swineford,
                ci_method = "wald", effect_sizes = FALSE)
```

```
fit2 <- mlmr_mv(cbind(t6_paragraph_comprehension, t9_word_meaning) ~
                  t5_general_information + t7_sentence,
                data = holzinger_swineford,
                ci_method = "wald", effect_sizes = FALSE)
anova(fit1, fit2)
```

`anova_within` *One Way Within-Subjects ANOVA With Sphericity Diagnostics and Corrections*

Description

Performs the univariate one-way within-subjects F test together with Mauchly's test of sphericity and the three standard ε-corrected p-values (Greenhouse-Geisser, Huynh-Feldt, and lower-bound). Returns everything in a single tidy `data.frame` so the user can decide which adjustment to report.

Usage

```
anova_within(x, id = NULL, time = NULL, outcome = NULL)
```

Arguments

`x`	Either an $n \times k$ numeric matrix or `data.frame` (rows = subjects, columns = repeated measurements); *or* a long-format `data.frame` together with `id`, `time`, and `outcome` column names.
`id`	Column name in `x` identifying the subject when `x` is in long format (`NULL` otherwise).
`time`	Column name in `x` identifying the within-subjects factor level when `x` is in long format (`NULL` otherwise).
`outcome`	Column name in `x` identifying the dependent variable when `x` is in long format (`NULL` otherwise).

Details

The unadjusted within-subjects F statistic is the same regardless of sphericity; corrections shrink the numerator and denominator degrees of freedom by a factor of $\hat{\varepsilon} \in [1/(k-1),\ 1]$, and the p-value is recomputed against the adjusted reference F distribution. When Mauchly's test rejects, prefer the Huynh-Feldt-corrected p-value (less conservative than Greenhouse-Geisser).

For multi-factor within-subjects designs or mixed designs, fit the model with `stats::aov(... + Error(id/within))` or with `lme4::lmer()` directly.

Value

A `data.frame` with one row per reported F test: `adjustment` (`"none"`, `"Greenhouse-Geisser"`, `"Huynh-Feldt"`, `"lower_bound"`), `F_value`, `df_1`, `df_2`, `p_value`, and `epsilon` (the correction factor used; `NA` for the unadjusted row). `attr(<output>, "mauchly")` contains the row from `mauchly_test`, and the partial η^2 is attached as `attr(<output>, "partial_eta_squared")`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 11.)

See Also

mauchly_test, epsilon_corrections, aov

Other within-subjects analysis: anova_within_two_way(), epsilon_corrections(), mauchly_test(), pairwise_within(), plot_trajectories_fitted()

Other hypothesis tests: adjusted_means(), ancova(), ci_dunnett(), ci_scheffe(), ci_tukey_kramer(), compare_cov_structures(), contrast_test(), correlations_test(), equivalence_r(), equivalence_smd(), factorial_anova(), manova_split_plot(), mauchly_test(), mixed_anova(), obrien_test(), pairwise_within(), randomization_test(), randomization_test_paired(), regions_of_significance(), simple_effects_AB(), summary_t_test(), welch_t()

Examples

```
# Simulated within-subjects data with no real effect.
set.seed(113)
Y <- matrix(rnorm(20 * 4), nrow = 20)
anova_within(Y)

# Built-in within-subjects example: nlme::Orthodont (distance ~ age).
res <- anova_within(nlme::Orthodont,
                    id = "Subject", time = "age", outcome = "distance")
res
attr(res, "mauchly")
attr(res, "partial_eta_squared")
```

---

anova_within_two_way *Two-Factor Within-Subjects ANOVA With Sphericity Adjustments*

Description

Computes the full two-factor within-subjects ANOVA table, main effects A, B, and the $A \times B$ interaction with each effect tested against its own residual stratum and with Greenhouse-Geisser, Huynh-Feldt, and lower-bound sphericity adjustments applied per effect. Returns a tidy long-form data.frame that composes with the rest of DMAR.

Usage

```
anova_within_two_way(data, outcome, factor_A, factor_B, subject)
```

Arguments

`data`	A `data.frame` in long format, with one row per subject-by-cell observation.
`outcome`	Character name of the response column in `data`.
`factor_A`	Character name of the first within-subjects factor.
`factor_B`	Character name of the second within-subjects factor.
`subject`	Character name of the subject-id column.

Details

Design. The two within-subjects factors *A* (with a levels) and *B* (with b levels) are fully crossed; every subject contributes $a \cdot b$ observations. Each of the three fixed effects is tested against its own subject-by-effect residual stratum:

- $F_A = MS_A / MS_{A:S}$
- $F_B = MS_B / MS_{B:S}$
- $F_{A:B} = MS_{A:B} / MS_{A:B:S}$

Sphericity. Each effect's univariate *F*-ratio assumes sphericity of its corresponding subject-by-effect residual covariance matrix. Three adjustments are reported per effect: Greenhouse-Geisser (Greenhouse & Geisser, 1959), Huynh-Feldt (Huynh & Feldt, 1976), and the lower bound $\epsilon = 1/df$, where *df* is the effect's numerator degrees of freedom; this is the smallest value ϵ can attain, reached under maximal departure from sphericity.

Subjects needed to estimate epsilon. The Greenhouse-Geisser epsilon for an effect with q numerator degrees of freedom is estimated from the sample covariance matrix of q orthonormal contrasts among the effect's cell means, a different matrix for each effect (Maxwell, Delaney, & Kelley, 2027, Chapters 11 and 12). That matrix has rank at most $n - 1$, so when $n - 1 < q$ it is necessarily singular; the same rank deficiency makes the multivariate approach to a within-subjects design mathematically impossible when $n < a$ (Maxwell, Delaney, & Kelley, 2027, Chapter 13). The Greenhouse-Geisser formula still returns a number in that case, but the number is an artifact of the rank deficiency rather than an estimate: it cannot exceed $(n - 1)/q$ no matter what the population epsilon is, even under exact sphericity, where the population value is 1. Rather than report a value the design cannot support, the function reports `NA` for the Greenhouse-Geisser and Huynh-Feldt rows of any effect with $n - 1 < q$ and issues a single warning naming the condition (`car::Anova` likewise declines to report the corrections for an effect whose error matrix is singular). The unadjusted row and the lower-bound row remain: the lower bound $1/q$ is Geisser and Greenhouse's a priori bound on epsilon, valid no matter how badly sphericity is violated, and it requires no estimate of the covariance matrix (Maxwell, Delaney, & Kelley, 2027, Chapter 11).

Per-effect partial η^2. Computed as $SS_{\text{effect}} / (SS_{\text{effect}} + SS_{\text{effect, error}})$ using the appropriate subject-by-effect residual sum of squares.

Balanced data assumed. The implementation assumes a fully balanced design (every subject observed once in every cell). When the design is unbalanced, the function errors and recommends a mixed-effects fit via `lmer`.

Sums of squares are unambiguous here. Because the design is balanced, the within-subjects factors are orthogonal and the Type I, Type II, and Type III sums of squares for each effect coincide. A Type toggle is therefore not meaningful, and the reported decomposition is unambiguous: the sum of squares attributed to each effect does not depend on the order in which terms enter the model (Maxwell, Delaney, & Kelley, 2027, Chapter 12).

Value

A `data.frame` with rows for each of the three effects (`A`, `B`, `A:B`) crossed with each sphericity adjustment (`none`, `Greenhouse-Geisser`, `Huynh-Feldt`, `lower_bound`). Columns: `effect`, `adjustment`, `F_value`, `df_1`, `df_2`, `p_value`, `epsilon`, `partial_eta_squared`. When an effect has too few subjects for its epsilon to be estimable ($n-1$ smaller than the effect's numerator degrees of freedom; see Details), the Greenhouse-Geisser and Huynh-Feldt rows for that effect carry `NA` in `epsilon`, `df_1`, `df_2`, and `p_value`, and a single warning names the condition; the unadjusted and lower-bound rows are unaffected.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Greenhouse, S. W., & Geisser, S. (1959). On methods in the analysis of profile data. *Psychometrika, 24*(2), 95–112.

Huynh, H., & Feldt, L. S. (1976). Estimation of the Box correction for degrees of freedom from sample data in randomized block and split-plot designs. *Journal of Educational Statistics, 1*(1), 69–82.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapters 11–13.)

See Also

`anova_within`, `mauchly_test`, `epsilon_corrections`

Other within-subjects analysis: `anova_within()`, `epsilon_corrections()`, `mauchly_test()`, `pairwise_within()`, `plot_trajectories_fitted()`

Examples

```
# 1. Balanced 3x2 within-subjects design simulated for illustration.
set.seed(113)
n_sub <- 12
grid  <- expand.grid(subject = factor(1:n_sub),
                     A = factor(c("a1", "a2", "a3")),
                     B = factor(c("b1", "b2")))
grid$y <- with(grid,
  3 * (A == "a2") + 1 * (A == "a3") +
  2 * (B == "b2") + 1 * ((A == "a3") & (B == "b2")) +
  rnorm(nrow(grid), 0, 1) +
  rep(rnorm(n_sub, 0, 1.5), times = 6))
anova_within_two_way(grid, outcome = "y", factor_A = "A",
                     factor_B = "B", subject = "subject")
```

`average_variance_extracted` *Average Variance Extracted (AVE)*

Description

The average variance extracted (AVE) is the mean proportion of indicator variance a factor accounts for, a standard convergent validity summary for a reflective measurement block in confirmatory factor analysis and structural equation modeling. With standardized loadings ℓ_j,

$$\mathrm{AVE} = \frac{1}{J}\sum_{j} \ell_j^2.$$

The quantity itself is elementary. In a model with cross-loadings, an item contributes its loading to the AVE of every factor it loads on; the same shared variance then counts toward each factor's summary, so compare AVE values across factors of such a model with that overlap in mind. Fornell and Larcker (1981) are credited for establishing it as a validity criterion: a construct shows convergent validity when its AVE reaches the conventional 0.50 (the construct explains at least half its indicators' variance), and the Fornell-Larcker discriminant criterion compares each construct's AVE with its squared correlations with the other constructs. AVE is closely related to composite reliability (omega); the modern complement on the discriminant side is `htmt`.

Usage

```
average_variance_extracted(
  fit = NULL,
  loadings = NULL,
  conf_level = 0.95,
  ci_method = c("none", "percentile"),
  B = 1000L,
  seed = NULL
)
```

Arguments

`fit`	Optional **lavaan** fit (for example from `cfa_1` or `lavaan::cfa`); the standardized loadings of every factor are extracted and one AVE is reported per factor.
`loadings`	Optional numeric vector of standardized loadings for a single block, as an alternative to `fit`. Supply exactly one of `fit` and `loadings`. No interval can be constructed from loadings alone (their sampling variability is not carried by the numbers), so `ci_method = "percentile"` requires `fit`.
`conf_level`	Confidence level for the bootstrap interval (default `0.95`); used when `ci_method = "percentile"`.
`ci_method`	Interval method: `"none"` (the default) or `"percentile"`. No closed-form interval for the AVE is in common use; the percentile bootstrap is the standard route

in the validity literature. The cases in the fitted data are resampled with replacement `B` times, the model is refit to each resample, and each factor's interval is the pair of empirical quantiles of its `B` AVE values (Efron & Tibshirani, 1993). Replications whose refit fails or does not converge are dropped, and the interval is computed from those that return a value; a single warning reports how many were dropped.

`B` Number of bootstrap replications when `ci_method = "percentile"` (default `1000`). The default is smaller than the package's usual `10000` because every replication refits the model; raise it for a reported analysis when time allows.

`seed` Optional integer seed for the bootstrap. The default `NULL` uses the current state of the random number generator; a supplied seed is set internally and the prior state restored on exit.

Details

The percentile bootstrap interval resamples the cases behind `fit` and refits the model once per replication, so its cost is `B` model fits. That refitting is why the examples below stop at the point estimates: even the smallest permitted `B = 100` runs for several seconds on the two-factor model there. To obtain the interval, pass `ci_method = "percentile"` together with a `seed`, as in `average_variance_extracted(fit, ci_method = "percentile", seed = 113)`, and keep the default `B = 1000` or more for a reported analysis.

Value

A `data.frame` (class `dmar_tbl`) with one row per factor: `factor` (label), `ave`, and `ci_lower` / `ci_upper` (the percentile bootstrap limits; `NA` when `ci_method = "none"`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, NY: Chapman & Hall/CRC.

Fornell, C., & Larcker, D. F. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research, 18*(1), 39–50.

See Also

`htmt` for discriminant validity; `reliability_omega` for the composite reliability of the same block (coefficient omega is what the composite-reliability literature computes); `cfa_1` to obtain the fit.

Other multivariate and latent variable methods: `bifactor_indices()`, `cfa_1()`, `cfa_2()`, `cfa_k()`, `ci_eigenvalue()`, `common_method_marker()`, `common_method_single_factor()`, `dmacs()`, `ecvi()`, `htmt()`, `irt_grm()`, `irt_information()`, `measurement_alignment()`, `measurement_invariance()`, `procrustes_phi()`, `simple_structure()`

Examples

```
# Directly from the standardized loadings a paper reports:
average_variance_extracted(loadings = c(.8, .7, .6))

# From a fitted model, one AVE per factor (requires lavaan).
data(holzinger_swineford)
fit <- lavaan::cfa(
  "verbal    =~ t6_paragraph_comprehension + t7_sentence +
                t9_word_meaning
   deduction =~ t20_deduction + t22_problem_reasoning +
                t23_series_completion",
  data = holzinger_swineford)
ave_tbl <- average_variance_extracted(fit)
ave_tbl

# The Fornell and Larcker (1981) discriminant criterion compares each
# factor's AVE with the squared correlation between the factors: a
# factor should account for more of its own indicators' variance than
# it shares with the other factor. Here the comparison favors verbal
# and goes against deduction, whose AVE falls below the shared
# variance. Fitting with cfa_k(..., output = "measurement") puts the
# AVE values and the latent correlations in one table.
lavaan::lavInspect(fit, "cor.lv")["verbal", "deduction"]^2

# The broom verbs: one row per factor.
generics::tidy(ave_tbl)
generics::glance(ave_tbl)
```

`bayes_independent_t` *Bayesian Independent-Samples* t *Analysis*

Description

The Bayesian counterpart of the two-sample (pooled-variance) t test. It reports the posterior of the standardized mean difference $\delta = (\mu_1 - \mu_2)/\sigma$ under the default Jeffreys-Zellner-Siow (JZS) prior, a Cauchy prior on δ with Jeffreys priors on the nuisance parameters (the variance), summarized by its median, mean, a credible interval, and the probability statement $P(\delta > 0 \mid \text{data})$. The JZS default Bayes factor (Rouder, Speckman, Sun, Morey, & Iverson, 2009) is also reported. See `bayes_one_sample_t` for the model, the package's interpretive stance, and the computational details (exact quadrature, no Monte Carlo error).

Usage

```
bayes_independent_t(
  x = NULL,
  y = NULL,
  mean_1 = NULL,
  sd_1 = NULL,
```

```
  n_1 = NULL,
  mean_2 = NULL,
  sd_2 = NULL,
  n_2 = NULL,
  prior_location = 0,
  prior_scale = sqrt(2)/2,
  prior_mean = NULL,
  prior_sd = NULL,
  conf_level = 0.95
)
```

Arguments

x, y	Numeric vectors: the observations of the two independent groups (δ is positive when x runs higher). Omit both to supply summary statistics instead.
mean_1, sd_1, n_1	Summary statistics of the first group: mean, standard deviation, and sample size. The Bayes factor depends on the data only through the *t* statistic and the sample sizes, so the summary form is exact, not an approximation. Supply either raw data or all six summary values, never both.
mean_2, sd_2, n_2	Summary statistics of the second group.
prior_location	Location of the Cauchy prior on δ. Defaults to 0, the JZS prior; a nonzero value centers the prior on an expected effect (Gronau, Ly, & Wagenmakers, 2020).
prior_scale	The way to adjust the prior. It is the scale (width) of the Cauchy prior on the standardized effect δ (the JZS prior), so a user who wants a more or less informative prior sets prior_scale: larger values say larger effects are plausible a priori, smaller values concentrate the prior near zero. The default $\sqrt{2}/2 \approx 0.707$ is the JZS "medium" prior. Fully custom or subjective priors beyond the Cauchy family are not supported by the **BayesFactor** engine.
prior_mean, prior_sd	Mean and standard deviation of a normal prior on δ, for prior beliefs stated as moments. Supplying them selects the normal prior; they cannot be combined with the Cauchy arguments. See the prior section of Details.
conf_level	Probability mass of the credible interval.

Details

The likelihood of the pooled-variance *t* statistic given δ is noncentral *t* with $df = n_1 + n_2 - 2$ and noncentrality $\delta\sqrt{n_1 n_2/(n_1 + n_2)}$; equal variances are assumed, as in the standard JZS development. The raw-scale rows transform the δ summaries through the pooled standard deviation.

Value

A data.frame (class dmar_tbl) with the same rows as bayes_one_sample_t, plus n_1 and n_2.

Specifying the prior. The default prior on the standardized effect δ is the JZS Cauchy centered at zero. Its prior_scale r is not a standard deviation: a Cauchy has no mean and no variance

(those integrals diverge), so beliefs stated as prior moments cannot be expressed through it. What the scale does fix is the quartiles: half the prior mass lies within $\pm r$ of the location, so the default $r = \sqrt{2}/2$ says a 50 percent prior bet that $|\delta| < 0.71$. A directional prior keeps the Cauchy and moves `prior_location` (Gronau, Ly, & Wagenmakers, 2020). A researcher who thinks in prior moments instead sets `prior_mean` and `prior_sd`, which use a normal prior with exactly those moments; the two families are exclusive.

The families are also linked by an exact identity: a Cauchy with location μ and scale r is a normal prior $N(\mu, r^2/z^2)$ whose z is standard normal, that is, a normal prior whose variance you are not sure of. Choosing the Cauchy is therefore choosing a normal prior with built-in doubt about its own width, which is why its tails are heavier and its Bayes factors more conservative. A normal matched to the Cauchy's interquartile range has `prior_sd = 1.4826 * prior_scale`. The full posterior of δ is returned in the `"posterior"` attribute as a data frame of `delta` and `density`, so any posterior probability, not only the reported ones, can be computed from it.

A standardized effect size enters through the summary form directly: an observed Cohen's d with group sizes n_1 and n_2 is `mean_1 = d, mean_2 = 0, sd_1 = 1, sd_2 = 1`, since d is the mean difference in pooled standard deviation units.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Gronau, Q. F., Ly, A., & Wagenmakers, E.-J. (2020). Informed Bayesian t-tests. *The American Statistician, 74*(2), 137–143. doi:10.1080/00031305.2018.1562983

Rouder, J. N., Speckman, P. L., Sun, D., Morey, R. D., & Iverson, G. (2009). Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic Bulletin & Review, 16*(2), 225–237. doi:10.3758/PBR.16.2.225

Jeffreys, H. (1961). *Theory of probability* (3rd ed.). Oxford University Press.

Zellner, A., & Siow, A. (1980). Posterior odds ratios for selected regression hypotheses. In J. M. Bernardo, M. H. DeGroot, D. V. Lindley, & A. F. M. Smith (Eds.), *Bayesian statistics: Proceedings of the First International Meeting* (pp. 585–603). University of Valencia Press.

See Also

`bayes_one_sample_t` and `bayes_paired_t` for the other designs; `ci_smd` and `welch_t` for the frequentist analyses of the same comparison.

Other Bayesian t analyses: `bayes_one_sample_t()`, `bayes_paired_t()`

Examples

```
set.seed(113)
g1 <- rnorm(35, 105, 15)
g2 <- rnorm(35, 100, 15)
bayes_independent_t(g1, g2)

# The probability that the effect is positive is read directly off the
# p_delta_positive row: a probability statement, not a p-value.
```

`bayes_one_sample_t` *Bayesian One-Sample* t *Analysis*

Description

The Bayesian counterpart of the one-sample *t* test. It reports the posterior distribution of the standardized effect $\delta = (\mu - \mu_0)/\sigma$ under the default Jeffreys-Zellner-Siow (JZS) prior, a Cauchy prior on δ with Jeffreys priors on the nuisance parameters (the variance), summarized by its median, mean, a credible interval, and the direct probability statement $P(\delta > 0 \mid \mathrm{data})$. The JZS default Bayes factor (Rouder, Speckman, Sun, Morey, & Iverson, 2009) is also reported.

Usage

```
bayes_one_sample_t(
  x = NULL,
  mu_0 = 0,
  mean = NULL,
  sd = NULL,
  n = NULL,
  prior_location = 0,
  prior_scale = sqrt(2)/2,
  prior_mean = NULL,
  prior_sd = NULL,
  conf_level = 0.95
)
```

Arguments

`x`	Numeric vector of observations. Omit to supply summary statistics instead.
`mu_0`	The comparison value for the mean under the point null (and the centering value for δ). Defaults to 0.
`mean`, `sd`, `n`	Summary statistics: the sample mean, standard deviation, and sample size. The Bayes factor depends on the data only through the *t* statistic and n, so the summary form is exact, not an approximation. Supply either `x` or all three summary values, never both.
`prior_location`	Location of the Cauchy prior on δ. Defaults to 0, the JZS prior; a nonzero value centers the prior on an expected effect (Gronau, Ly, & Wagenmakers, 2020).
`prior_scale`	The way to adjust the prior. It is the scale (width) r of the Cauchy prior on the standardized effect δ (the JZS prior), so a user who wants a more or less informative prior sets `prior_scale`: larger values say larger effects are plausible a priori, smaller values concentrate the prior near zero. The default $\sqrt{2}/2 \approx 0.707$ is the JZS "medium" prior. Fully custom or subjective priors beyond the Cauchy family are not supported by the **BayesFactor** engine.
`prior_mean`, `prior_sd`	Mean and standard deviation of a normal prior on δ, for prior beliefs stated as moments. Supplying them selects the normal prior; they cannot be combined with the Cauchy arguments. See the prior section of Details.

`conf_level` Probability mass of the (central) credible interval. Defaults to 0.95.

Details

The posterior is a *probability statement* about the parameter given the model, the prior, and the data, and that is how these functions are meant to be read. A Bayes factor is a different kind of claim, a comparison of how well two models predicted the data, and it leans harder on the prior; it is reported because it may be helpful for some questions. Neither replaces the estimation-first habits of the rest of the package; `ci_sm` and `ci_smd` remain the frequentist complements.

With a Jeffreys prior on (μ_0, σ^2) and $\delta \sim \text{Cauchy}(0, r)$, all inference flows through the observed t statistic, whose likelihood given δ is noncentral t with noncentrality $\delta\sqrt{n}$. The posterior of δ is computed by quadrature (no Monte Carlo error) and the Bayes factor by the one-dimensional integral of that likelihood against the Cauchy prior, the exact JZS form. The raw-scale rows transform the δ summaries through the sample standard deviation (a plug-in, as is conventional for reporting).

Value

A `data.frame` (class `dmar_tbl`) with the posterior summaries of δ (`delta_posterior_median`, `delta_posterior_mean`, `delta_lower`, `delta_upper`, `p_delta_positive`), the same summaries mapped to the raw mean difference scale (`raw_*`), the Bayes factors (`bf_10`, `bf_01`), the observed `t` and `df`, the `prior_scale`, and `n`.

Specifying the prior. The default prior on the standardized effect δ is the JZS Cauchy centered at zero. Its `prior_scale` r is not a standard deviation: a Cauchy has no mean and no variance (those integrals diverge), so beliefs stated as prior moments cannot be expressed through it. What the scale does fix is the quartiles: half the prior mass lies within $\pm r$ of the location, so the default $r = \sqrt{2}/2$ says a 50 percent prior bet that $|\delta| < 0.71$. A directional prior keeps the Cauchy and moves `prior_location` (Gronau, Ly, & Wagenmakers, 2020). A researcher who thinks in prior moments instead sets `prior_mean` and `prior_sd`, which use a normal prior with exactly those moments; the two families are exclusive.

The families are also linked by an exact identity: a Cauchy with location μ and scale r is a normal prior $N(\mu, r^2/z^2)$ whose z is standard normal, that is, a normal prior whose variance you are not sure of. Choosing the Cauchy is therefore choosing a normal prior with built-in doubt about its own width, which is why its tails are heavier and its Bayes factors more conservative. A normal matched to the Cauchy's interquartile range has `prior_sd = 1.4826 * prior_scale`. The full posterior of δ is returned in the `"posterior"` attribute as a data frame of `delta` and `density`, so any posterior probability, not only the reported ones, can be computed from it.

A standardized effect size enters through the summary form directly: an observed *d* relative to `mu_0 = 0` is `mean = d, sd = 1`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Gronau, Q. F., Ly, A., & Wagenmakers, E.-J. (2020). Informed Bayesian t-tests. *The American Statistician, 74*(2), 137–143. doi:10.1080/00031305.2018.1562983

Rouder, J. N., Speckman, P. L., Sun, D., Morey, R. D., & Iverson, G. (2009). Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic Bulletin & Review, 16*(2), 225–237. doi:10.3758/PBR.16.2.225

Jeffreys, H. (1961). *Theory of probability* (3rd ed.). Oxford University Press.

Zellner, A., & Siow, A. (1980). Posterior odds ratios for selected regression hypotheses. In J. M. Bernardo, M. H. DeGroot, D. V. Lindley, & A. F. M. Smith (Eds.), *Bayesian statistics: Proceedings of the First International Meeting* (pp. 585–603). University of Valencia Press.

See Also

`bayes_paired_t` and `bayes_independent_t` for the two-sample designs; `ci_sm` for the frequentist standardized mean.

Other Bayesian t analyses: `bayes_independent_t()`, `bayes_paired_t()`

Examples

```
set.seed(113)
x <- rnorm(40, mean = 0.4, sd = 1)
bayes_one_sample_t(x)

# Against a nonzero comparison value, with a wider prior.
bayes_one_sample_t(x, mu_0 = 0.1, prior_scale = 1)
```

`bayes_paired_t` *Bayesian Paired-Samples* t *Analysis*

Description

The Bayesian counterpart of the paired *t* test, the analysis of `bayes_one_sample_t` applied to the within-pair differences. It reports the posterior of the standardized difference $\delta = \mu_D/\sigma_D$ under the default Jeffreys-Zellner-Siow (JZS) prior, a Cauchy prior on δ with Jeffreys priors on the nuisance parameters (the variance), summarized by its median, mean, a credible interval, and the probability statement $P(\delta > 0 \mid \text{data})$. The JZS default Bayes factor (Rouder, Speckman, Sun, Morey, & Iverson, 2009) is also reported. See `bayes_one_sample_t` for the model, the package's interpretive stance, and the computational details (exact quadrature, no Monte Carlo error).

Usage

```
bayes_paired_t(
  x = NULL,
  y = NULL,
  mean_diff = NULL,
  sd_diff = NULL,
  n = NULL,
  prior_location = 0,
  prior_scale = sqrt(2)/2,
```

```
  prior_mean = NULL,
  prior_sd = NULL,
  conf_level = 0.95
)
```

Arguments

x, y	Numeric vectors of paired observations, the same length, in matching order. The analysis is of x - y. Omit both to supply summary statistics of the differences instead.
mean_diff, sd_diff, n	Summary statistics of the paired differences: their mean, their standard deviation, and the number of pairs. These are the quantities a paper's paired t test reports. Supply either raw data or all three summary values, never both.
prior_location	Location of the Cauchy prior on δ. Defaults to 0, the JZS prior; a nonzero value centers the prior on an expected effect (Gronau, Ly, & Wagenmakers, 2020).
prior_scale	The way to adjust the prior. It is the scale (width) of the Cauchy prior on the standardized effect δ (the JZS prior), so a user who wants a more or less informative prior sets prior_scale: larger values say larger effects are plausible a priori, smaller values concentrate the prior near zero. The default $\sqrt{2}/2 \approx 0.707$ is the JZS "medium" prior. Fully custom or subjective priors beyond the Cauchy family are not supported by the **BayesFactor** engine.
prior_mean, prior_sd	Mean and standard deviation of a normal prior on δ, for prior beliefs stated as moments. Supplying them selects the normal prior; they cannot be combined with the Cauchy arguments. See the prior section of Details.
conf_level	Probability mass of the credible interval.

Value

A data.frame (class dmar_tbl) with the same rows as bayes_one_sample_t, where δ is the standardized within-pair difference and the raw rows are on the difference scale; n is the number of pairs.

Specifying the prior. The default prior on the standardized effect δ is the JZS Cauchy centered at zero. Its prior_scale r is not a standard deviation: a Cauchy has no mean and no variance (those integrals diverge), so beliefs stated as prior moments cannot be expressed through it. What the scale does fix is the quartiles: half the prior mass lies within $\pm r$ of the location, so the default $r = \sqrt{2}/2$ says a 50 percent prior bet that $|\delta| < 0.71$. A directional prior keeps the Cauchy and moves prior_location (Gronau, Ly, & Wagenmakers, 2020). A researcher who thinks in prior moments instead sets prior_mean and prior_sd, which use a normal prior with exactly those moments; the two families are exclusive.

The families are also linked by an exact identity: a Cauchy with location μ and scale r is a normal prior $N(\mu, r^2/z^2)$ whose z is standard normal, that is, a normal prior whose variance you are not sure of. Choosing the Cauchy is therefore choosing a normal prior with built-in doubt about its own width, which is why its tails are heavier and its Bayes factors more conservative. A normal matched to the Cauchy's interquartile range has prior_sd = 1.4826 * prior_scale. The full posterior of δ is returned in the "posterior" attribute as a data frame of delta and density, so any posterior probability, not only the reported ones, can be computed from it.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Gronau, Q. F., Ly, A., & Wagenmakers, E.-J. (2020). Informed Bayesian t-tests. *The American Statistician, 74*(2), 137–143. doi:10.1080/00031305.2018.1562983

Rouder, J. N., Speckman, P. L., Sun, D., Morey, R. D., & Iverson, G. (2009). Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic Bulletin & Review, 16*(2), 225–237. doi:10.3758/PBR.16.2.225

Jeffreys, H. (1961). *Theory of probability* (3rd ed.). Oxford University Press.

Zellner, A., & Siow, A. (1980). Posterior odds ratios for selected regression hypotheses. In J. M. Bernardo, M. H. DeGroot, D. V. Lindley, & A. F. M. Smith (Eds.), *Bayesian statistics: Proceedings of the First International Meeting* (pp. 585–603). University of Valencia Press.

See Also

bayes_one_sample_t for the model and stance; bayes_independent_t for unpaired groups; probability_of_superiorit and randomization_test_paired for other paired analyses.

Other Bayesian t analyses: bayes_independent_t(), bayes_one_sample_t()

Examples

```
set.seed(113)
before <- rnorm(30, 100, 12)
after  <- before + rnorm(30, 3, 6)
bayes_paired_t(after, before)
```

---

bessel_errors — *Bessel's (1818) Grouped Frequency Distribution of Bradley's Astronomical Observation Errors*

Description

The nine-bin grouped frequency distribution that Friedrich Wilhelm Bessel published in 1818 for the absolute errors of 300 stellar position observations made by British Astronomer Royal James Bradley at the Greenwich Observatory between 1750 and 1762. Bessel compared the empirical distribution of these errors to the normal distribution, providing one of the early empirical demonstrations that observational errors are approximately normally distributed, a position Gauss had developed on theoretical grounds a decade earlier. The data are reproduced from Maxwell, Delaney, and Kelley (2027, *Designing Experiments and Analyzing Data: A Model Comparison Perspective*, 4th ed., Routledge), Table 1.4.

Usage

```
bessel_errors
```

Format

A data frame with 9 observations on 6 variables, one row per bin of the grouped frequency distribution. The error magnitudes are in seconds of arc.

`bin` Integer bin index, 1 through 9.

`lower` Lower edge of the bin (inclusive), in seconds of arc.

`upper` Upper edge of the bin (exclusive), in seconds of arc.

`midpoint` Bin midpoint, `(lower + upper) / 2`, in seconds of arc. The conventional plug-in value when approximating moments from a grouped frequency distribution.

`observed` Empirical frequency: the number of Bradley's 300 absolute errors that fell in the bin.

`expected` Expected frequency under a normal distribution with mean 0 and standard deviation approximately 0.22 seconds of arc (a least squares fit to the expected counts gives 0.216). These are Bessel's own normal-model expectations as reproduced in Maxwell, Delaney, and Kelley (2027). Both the observed and expected columns sum to 300.

Details

This data set ships in the original grouped form Bessel reported. The 300 individual error values are not available; what Bessel published, and what is reproduced here, is the 9-bin frequency distribution. Computations that require the underlying continuous values must either be approximated from the bin midpoints (the usual weighted-moments approach, illustrated in the examples) or estimated parametrically by assuming a distributional form within each bin.

Historical context. Friedrich Wilhelm Bessel (1784–1846) was a German astronomer and mathematician best known to statisticians for the Bessel correction (`n - 1` in the unbiased variance estimator) and the Bessel functions. The 1818 monograph that contains this frequency distribution is part of a much larger effort to produce a reference catalog of stellar positions, the *Fundamenta astronomiae*, derived from the observations of James Bradley (1693–1762), the third Astronomer Royal of Britain and a pioneering observational astronomer. Bradley's position measurements were the most accurate of his era; the observational errors are small (most under 0.5 seconds of arc) and approximately normally distributed.

Why this data set matters for measurement and analysis. Bessel's 1818 comparison is one of the earliest empirical demonstrations that observational error is approximately normal, complementing the theoretical case Gauss had made on independent grounds. It is also a clean worked example for the approximation of moments from grouped frequency data when only binned counts (rather than individual observations) are available, a common situation in published reports.

Approximating moments from grouped data. When the underlying continuous values are unavailable, the standard approach is to plug the bin midpoints in for the unknown individual values and form a weighted mean and weighted variance using the bin frequencies as weights. The frequency-weighted mean and variance are

$$\bar{x}_w = \frac{\sum_k f_k m_k}{\sum_k f_k}, \qquad s_w^2 = \frac{\sum_k f_k (m_k - \bar{x}_w)^2}{(\sum_k f_k) - 1}$$

where f_k is the bin frequency and m_k is the bin midpoint. The examples below compute both, on the observed frequencies and on Bessel's normal-model expected frequencies. The two are close, consistent with Bessel's conclusion that the empirical and theoretical distributions agree.

Author(s)

Ken Kelley

Source

Bessel, F. W. (1818). *Fundamenta astronomiae pro anno MDCCLV deducta ex observationibus viri incomparabilis James Bradley in specula astronomica Grenovicensi per annos 1750–1762 institutis* [Foundations of astronomy for the year 1755, deduced from the observations of the incomparable man James Bradley at the Greenwich astronomical observatory during 1750–1762]. Friedrich Nicolovius.

Reproduced in Maxwell, Delaney, and Kelley (2027), Table 1.4.

References

Bessel, F. W. (1818). *Fundamenta astronomiae pro anno MDCCLV deducta ex observationibus viri incomparabilis James Bradley in specula astronomica Grenovicensi per annos 1750–1762 institutis*. Friedrich Nicolovius.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 1, Section 1.4.5.2, on the historical and empirical basis for the normal distribution.)

Kelley, K. (2026). *DMAR: Methods for the design, measurement, and analysis of human-centered outcomes in R* [R package]. https://github.com/yelleKneK/DMAR

Stigler, S. M. (1986). *The history of statistics: The measurement of uncertainty before 1900*. Belknap Press of Harvard University Press. (See Chapter 5 on the development of the normal distribution and the role of astronomical errors.)

Examples

```
data(bessel_errors)
bessel_errors

# Total frequencies (each column should sum to 300).
colSums(bessel_errors[, c("observed", "expected")])

# Weighted mean of the absolute error, approximating the
# individual observations by the bin midpoints. Because every
# bin lower edge is at or above zero, this is the mean
# absolute error rather than the mean error itself.
wmean_obs <- with(bessel_errors,
                  sum(observed * midpoint) / sum(observed))
wmean_exp <- with(bessel_errors,
                  sum(expected * midpoint) / sum(expected))
c(observed = wmean_obs, expected_under_normal = wmean_exp)

# Weighted variance and standard deviation of the absolute
# error, using bin midpoints as plug-in values for the
# individual observations.
wvar_obs <- with(bessel_errors,
                 sum(observed * (midpoint - wmean_obs)^2) /
```

```
                (sum(observed) - 1))
c(weighted_variance = wvar_obs,
  weighted_sd       = sqrt(wvar_obs))

# Side-by-side bar plot of observed and expected counts.
# Bessel's normal-model expectation tracks the empirical
# distribution closely except in the long right tail, where
# Bradley's three largest errors (counts 3, 1, 1) exceed what
# the normal model predicts.
op <- par(mar = c(5, 4, 4, 2))
barplot(rbind(bessel_errors$observed, bessel_errors$expected),
        beside    = TRUE,
        names.arg = sprintf("%.1f-%.1f",
                            bessel_errors$lower,
                            bessel_errors$upper),
        legend.text = c("Observed (Bradley)",
                        "Expected under normal model"),
        args.legend = list(x = "topright", bty = "n"),
        xlab = "Absolute error (seconds of arc)",
        ylab = "Frequency",
        main = "Bessel (1818) on Bradley's 300 stellar positions")
par(op)
```

`bifactor_indices` *Bifactor Model Dimensionality and Reliability Indices*

Description

Computes the indices used to judge whether a multidimensional scale is nonetheless unidimensional enough to score as a single total (Rodriguez, Reise, and Haviland, 2016): the explained common variance (ECV), coefficient omega and omega hierarchical (omega_H) for the general factor, omega hierarchical subscale (omega_HS) for each group factor, the percentage of uncontaminated correlations (PUC), and coefficient H, the construct reliability (maximal reliability) of Hancock and Mueller (2001). The input is a fitted bifactor model in which one general factor loads on every item and each item loads on exactly one orthogonal group factor.

Usage

```
bifactor_indices(fit, general = NULL)
```

Arguments

`fit`	A fitted bifactor **lavaan** model: one general factor on all items plus orthogonal group factors, each item on one group factor. The factors must be orthogonal (the bifactor specification).
`general`	Optional name of the general factor. When `NULL` (default) the general factor is detected as the one that loads on every item.

Details

Let λ_i^g be item i's standardized loading on the general factor, λ_i^s its loading on its group factor, and θ_i its standardized residual variance. With orthogonal factors the overall ECV is $\sum_i(\lambda_i^g)^2 / \sum_i[(\lambda_i^g)^2 + (\lambda_i^s)^2]$; omega and omega_H share the total-score variance $(\sum_i \lambda_i^g)^2 + \sum_g(\sum_{i \in g} \lambda_i^s)^2 + \sum_i \theta_i$ as denominator, with the general part $(\sum_i \lambda_i^g)^2$ in the numerator of omega_H. Each group factor's omega_HS uses the analogous numerator and that subscale's own total variance. PUC is one minus the share of item pairs that fall within the same group factor. Coefficient H is computed from $\sum \lambda^2/(1 - \lambda^2)$ on the relevant loadings (see `reliability_H`). High ECV and PUC with a high omega_H support scoring a single total; substantial subscale omega_HS argues for reporting subscales as well.

A bifactor model that is over-parameterized for the data frequently yields an improper (Heywood) solution – a standardized loading outside $[-1, 1]$ or a negative residual variance – whose indices are not trustworthy. When that happens the indices are still returned but a warning is issued and the `"improper"` attribute is set to `TRUE`.

Value

A `data.frame` (class `dmar_tbl`) with one row per factor (the general factor first, then each group factor) and columns `factor`, `ECV`, `omega`, `omega_H`, `omega_HS`, `PUC`, and `H`. Quantities that do not apply to a row are `NA` (for example omega_H and PUC on a group factor). The `"improper"` attribute flags a Heywood solution.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Hancock, G. R., & Mueller, R. O. (2001). Rethinking construct reliability within latent variable systems. In R. Cudeck, S. du Toit, & D. Sörbom (Eds.), *Structural equation modeling: Present and future* (pp. 195–216). Scientific Software International.

Reise, S. P. (2012). The rediscovery of bifactor measurement models. *Multivariate Behavioral Research, 47*(5), 667–696. doi:10.1080/00273171.2012.715555

Rodriguez, A., Reise, S. P., & Haviland, M. G. (2016). Evaluating bifactor models: Calculating and interpreting statistical indices. *Psychological Methods, 21*(2), 137–150.

See Also

`reliability_omega`, `reliability_H`.

Other multivariate and latent variable methods: `average_variance_extracted()`, `cfa_1()`, `cfa_2()`, `cfa_k()`, `ci_eigenvalue()`, `common_method_marker()`, `common_method_single_factor()`, `dmacs()`, `ecvi()`, `htmt()`, `irt_grm()`, `irt_information()`, `measurement_alignment()`, `measurement_invariance()`, `procrustes_phi()`, `simple_structure()`

Examples

```
# Nine items: one general factor and three orthogonal group factors.
set.seed(113)
n <- 600
g <- rnorm(n); grp <- list(rnorm(n), rnorm(n), rnorm(n))
X <- vapply(1:9, function(i)
  0.5 * g + 0.5 * grp[[ceiling(i / 3)]] + sqrt(0.5) * rnorm(n), numeric(n))
colnames(X) <- paste0("x", 1:9)
model <- "g  =~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9
          f1 =~ x1 + x2 + x3
          f2 =~ x4 + x5 + x6
          f3 =~ x7 + x8 + x9"
fit <- lavaan::cfa(model, data = as.data.frame(X),
                   orthogonal = TRUE, std.lv = TRUE)
bifactor_indices(fit)
```

---

bryant_paulson *The Bryant-Paulson Generalized Studentized Range Distribution*

Description

Distribution function (`pbryant_paulson`), quantile/critical-value function (`qbryant_paulson`), and density (`dbryant_paulson`) for the Bryant-Paulson generalized studentized range, the sampling distribution of the studentized range of covariate-*adjusted* means in the analysis of covariance (ANCOVA) when the covariate(s) are *random*. These are the analysis-of-covariance analogues of `ptukey` / `qtukey` and supply the critical values needed for Tukey-Kramer-type simultaneous confidence intervals on (and tests of) contrasts of adjusted means.

Usage

```
pbryant_paulson(q, num_covariates, num_groups, df, lower_tail = TRUE, ...)

qbryant_paulson(prob, num_covariates, num_groups, df, lower_tail = TRUE, ...)

dbryant_paulson(q, num_covariates, num_groups, df, ...)
```

Arguments

`q`	Vector of quantiles (values of the generalized studentized range statistic).
`num_covariates`	The number of random covariates, p ($p \geq 0$). With `num_covariates = 0` the distribution is exactly the ordinary studentized range and the functions reduce to `ptukey` / `qtukey`.
`num_groups`	The number of groups (treatments) being compared, k ($k \geq 2$); this is the "sample size" of the range.
`df`	The error degrees of freedom of the ANCOVA model, ν. In a one-way ANCOVA with N total observations, k groups, and p covariates, $\nu = N - k - p$.

`lower_tail`	Logical; if `TRUE` (default) probabilities are $P(Q \le q)$, otherwise $P(Q > q)$.
`...`	Additional arguments (currently unused; for extensibility).
`prob`	Vector of probabilities. For `qbryant_paulson` this is the cumulative probability (e.g., `0.95` returns the upper 5% critical value).

Details

The statistic. In a balanced ANCOVA with k groups and p random covariates, let $\hat{\theta}_i$ be the adjusted group means and $\hat{\sigma}_{y|x}$ the square root of the ANCOVA error mean square (on ν degrees of freedom). The Bryant-Paulson statistic is the studentized range of the adjusted means,

$$Q \;=\; \frac{\max_i \hat{\theta}_i - \min_i \hat{\theta}_i}{\hat{\sigma}_{y|x}\sqrt{K_1 - K_2}},$$

where $K_1 - K_2$ is the design constant that scales the variance of a single adjusted mean (for a one-way design with n per group, $K_1 - K_2 = 1/n$). Crucially, the studentizer uses only this "between-only" standard error: the extra sampling variability induced by having to *estimate* the covariate adjustment from random covariates is carried by the distribution of Q itself, not by a per-comparison standard-error correction. This is what distinguishes the procedure from naively applying Tukey's method to adjusted means.

The distribution. Bryant and Paulson (1976) give the exact CDF of Q_p in their Equation (17), a single integral over a variable that combines the χ^2_ν error estimate with a random covariate-shrinkage factor δ that, by their Equations (11)–(12), has a $\mathrm{Beta}((\nu+1)/2,\, p/2)$ distribution. Carrying out the error integral with the studentized-range routine `ptukey` reduces Equation (17) to the equivalent one-dimensional form

$$P(Q_p \le q) \;=\; \int_0^1 \mathrm{ptukey}\Big(q\sqrt{\delta};\, k,\, \nu\Big)\; f_{\mathrm{Beta}}\big(\delta;\, \tfrac{\nu+1}{2},\, \tfrac{p}{2}\big)\, d\delta,$$

which this package evaluates (the reduction is exact; see the source-code comments in '`R/bryant_paulson.R`' for the one-line derivation from Bryant and Paulson's p. 634 conditioning argument). When $p = 0$ the factor δ degenerates at 1 and Q_p is exactly the ordinary studentized range (Bryant and Paulson, 1976, Sec. 1), so the code short-circuits to `ptukey`. The integral is evaluated with `integrate` after the change of variables $\delta = 1 - u^2$, which removes the endpoint singularity of the Beta weight at $\delta = 1$ when $p = 1$ and makes the quadrature converge in a few subdivisions; `qbryant_paulson` inverts it with `uniroot`. Bryant and Bruvold (1980) later showed the same distribution and critical values remain valid when the covariates are *not* identically distributed across groups (their grouped-covariate model, Eq. 1.3), and added the Duncan multiple-range extension.

Accuracy at small df. The error integral is carried out with `ptukey` for $\nu \ge 7$, where it is accurate to about 10^{-9}. For $\nu < 7$ `ptukey`'s algorithm loses accuracy (at $\nu = 3$, $k = 20$ its probability error reaches $\approx 3 \times 10^{-4}$, enough to move the critical value by about 0.2, and it is larger at $\nu = 2$), so the studentized-range distribution is instead evaluated directly, without `ptukey`, by integrating the probability integral of the range against the χ^2_ν error density. The small-ν path costs a fraction of a second.

Validation. The implementation reproduces Bryant and Paulson's (1976) Table 1 exactly, to the two decimal places tabled, over the whole of its range: both tail areas ($\alpha = .05$ and $\alpha = .01$), all three covariate counts ($p = 1, 2, 3$), every tabled number of groups ($k = 2, \ldots, 8, 10, 12, 16, 20$), and every tabled error degrees of freedom ($\nu = 2, \ldots, 8, 10, 12, 14, 16, 18, 20, 24, 30, 40, 60, 120$),

which is 1188 critical values in all. The corners of the table are included: $q_{.01;\,1,2,2} = 19.09$, $q_{.01;\,3,20,2} = 73.01$, $q_{.01;\,3,20,3} = 33.13$, and $q_{.05;\,1,6,14} = 4.83$ (the value used in the Bryant and Bruvold, 1980 worked example). Two of the 1188 entries, $q_{.01;\,2,8,3}$ and $q_{.01;\,2,20,4}$, have exact values of 23.165013 and 19.745008, each roughly 10^{-5} above the 23.165 and 19.745 half-way points, so they round to 23.17 and 19.75; the 1976 table rounds them down, to 23.16 and 19.74. Both values were confirmed to fourteen significant figures by two independent high-order quadrature engines that share no code with the package implementation. A large-scale simulation of the Bryant and Paulson statistic confirms the computed values independently, and the Bryant and Bruvold (1980) Table 2 Duncan ranges are reproduced as well. See the package tests.

Value

Numeric vectors. `pbryant_paulson` returns cumulative (or upper-tail) probabilities, `dbryant_paulson` returns density values, and `qbryant_paulson` returns critical values (quantiles) of the generalized studentized range. Results are recycled to the length of the longest of `q`/`prob` and the parameter arguments.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bryant, J. L., & Paulson, A. S. (1976). An extension of Tukey's method of multiple comparisons to experimental designs with random concomitant variables. *Biometrika, 63*, 631–638. doi:10.1093/biomet/63.3.631

Bryant, J. L., & Bruvold, N. T. (1980). Multiple comparison procedures in the analysis of covariance. *Journal of the American Statistical Association, 75*(372), 874–880. doi:10.2307/2287175

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9.)

See Also

`ci_c_ancova_bp` for the simultaneous confidence intervals these critical values produce; `ptukey` and `qtukey` for the ordinary (fixed-covariate or no-covariate) studentized range.

Examples

```
# Critical value from the worked example of Bryant and Bruvold (1980):
# k = 6 panels, p = 1 covariate, nu = 14 error df, alpha = .05. The quantile
# is found by inverting the distribution function with uniroot; it is 4.83,
# the entry in Table 1 of Bryant and Paulson (1976).
qbryant_paulson(0.95, num_covariates = 1, num_groups = 6, df = 14)

# The ordinary Tukey value (ignoring that the covariate is random and
# estimated) is smaller, 4.64, so it yields intervals that are too narrow:
qtukey(0.95, nmeans = 6, df = 14)

# How much too narrow: the Bryant-Paulson area beyond the Tukey value is the
# familywise error rate Tukey's method actually delivers in this design.
```

```
# With no covariate the two distributions coincide, so the area is .0499,
# the nominal .05 up to the rounding of 4.64 itself. Each additional random
# covariate carries more estimation uncertainty, stretches the distribution
# to the right, and pushes the rate up, to .063, .076, and .091.
pbryant_paulson(4.64, num_covariates = 0:3, num_groups = 6, df = 14,
                lower_tail = FALSE)

# The p = 0 entry above is exactly the ordinary studentized range:
ptukey(4.64, nmeans = 6, df = 14, lower.tail = FALSE)
```

`cfa_1` *One Factor Confirmatory Factor Analysis Model*

Description

Fits a single factor (congeneric by default) confirmatory factor analysis model to raw item data or a sample covariance matrix. This is the one factor special case of `cfa_k`: the function is a convenience wrapper that only requires the data (and, when the data hold more than the items, a vector of item names), builds the one factor specification, and forwards everything else to `cfa_k()`. The factor is named `f1`, so the rows of the returned table are `lambda_f1_1`, `lambda_f1_2`, ..., `psi_f1_1`, ..., and `omega_f1`, exactly as a one factor `cfa_k()` call would report them (the `syntax` column names the item behind each number).

Usage

```
cfa_1(
  data = NULL,
  items = NULL,
  S = NULL,
  N = NULL,
  equal_loading = FALSE,
  equal_error = FALSE,
  estimator = "ML",
  missing = "listwise",
  se = "standard",
  conf_level = 0.95,
  output = c("verbose", "measurement", "summary", "standardized", "fit"),
  ...
)
```

Arguments

`data`	A raw data matrix or data frame, rows are respondents and columns include the items. A matrix without column names is given the names `y1`, `y2`, ... Supply exactly one of `data` or `S`.

`items` Character vector naming the items of the factor (three or more; two are accepted with `equal_loading = TRUE`, the just identified tau-equivalent case). The default `NULL` uses every column of `data` (or of `S`), so a data set that holds only the items needs no `items` at all.

`S` A symmetric covariance matrix of the items; `N` is then required. Dimnames are optional here: a matrix without them is given the item names `y1`, `y2`, ... Supply exactly one of `data` or `S`.

`N` Total sample size. Required with `S`; ignored (inferred from the rows) with `data`.

`equal_loading` Logical, or a named logical vector with one element per factor. `TRUE` constrains the loadings within a factor to a single value (across factors nothing is equated). Defaults to `FALSE`.

`equal_error` Logical, or a named logical vector with one element per factor. `TRUE` constrains the error variances within a factor to a single value. Defaults to `FALSE`.

`estimator` Character; estimator passed to **lavaan**. Must be one of `"ML"` (default; maximum likelihood, fully efficient under multivariate normality), `"MLR"` (robust maximum likelihood: maximum likelihood estimates with standard errors and test statistic corrected for nonnormality; Satorra & Bentler, 1994), `"WLS"` (the asymptotic distribution free estimator of Browne, 1984; raw data and a large sample required), `"WLSMV"` (diagonally weighted least squares with mean- and variance-adjusted test statistic; Muthén, 1984; Muthén, du Toit, & Spisic, 1997; the standard choice for ordered categorical items, and what `ordered` switches to), or `"GLS"` (generalized least squares; Browne, 1974). With a robust estimator the reported fit indices are the robust versions.

`missing` Character; missing-data handling passed to **lavaan** when raw data are supplied. Common values are `"listwise"` (default, listwise deletion) and `"ml"` (full information maximum likelihood). Ignored with `S`. With `ordered` items, `"ml"`/`"fiml"` are not available; use `"pairwise"` or `"listwise"`.

`se` Standard error type passed to **lavaan**; see `cfa`. Common values are `"standard"` (default), `"robust.sem"` (with `estimator = "MLR"`), and `"none"` (point estimates only; fastest).

`conf_level` Confidence level for the parameter confidence intervals, including the delta method intervals for omega, AVE, and *H*. Defaults to 0.95. The RMSEA interval is a separate convention (see Details).

`output` Format of the returned object:

`"verbose"` (default) Parameter estimates with confidence intervals, the per-factor defined parameters (`loading_sum`, `error_sum`, `omega`, `ave`, `H`), and fit information.

`"measurement"` The measurement-property rows only: per factor `omega`, `ave`, and `H` (with delta method standard errors and confidence intervals), the latent correlation `phi` for every factor pair (with its confidence interval), and, for raw data, the heterotrait-monotrait ratio `htmt` for every factor pair via `htmt`.

`"summary"` The raw `summary()` output from **lavaan** (not a data frame).

`"standardized"` The standardized parameter estimates from `lavaan::standardizedSolution()`.

"fit" The raw **lavaan** fit object. Escape hatch for direct lavaan access, including likelihood ratio tests between two cfa_k() fits via lavaan::lavTestLRT().

... Additional arguments forwarded to cfa_k and, through it, to lavaan.

Details

The model is identified by fixing the factor variance to 1 and estimating every loading. equal_loading and equal_error impose the classical measurement structures on the single factor, and the header of the printed table names the structure implied by the constraints. For the composite reliability coefficient with the observed total variance in the denominator, use reliability_omega with denominator = "observed"; for the model implied omega, the omega_f1 row of this function's output and reliability_omega agree.

Ordered categorical items are not supported here; use cfa_k, whose ordered argument fits WLSMV with the theta parameterization and reports the Green and Yang (2009) categorical sum score omega, or reliability_omega_categorical.

Value

The value of the corresponding cfa_k call: a data.frame (classes dmar_cfa_k, dmar_tbl) with one row per parameter (estimate, se, z_value, p_value, ci_lower, ci_upper) followed by the fit rows, or the alternative shapes selected by output (see ?cfa_k).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Green, S. B., & Yang, Y. (2009). Reliability of summed item scores using structural equation modeling: An alternative to coefficient alpha. *Psychometrika, 74*(1), 155–167. doi:10.1007/s11336008-90993

Kelley, K., & Pornprasertmanit, S. (2016). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for composite measures. *Psychological Methods, 21*(1), 69–92. doi:10.1037/a0040086

McDonald, R. P. (1999). *Test theory: A unified treatment*. Lawrence Erlbaum Associates.

See Also

cfa_k for the general function this wraps (factor analysis with any number of factors, intercept constraints, ordered categorical items, and the measurement output); cfa_2 for the two factor wrapper; reliability_omega for coefficient omega with confidence intervals.

Other multivariate and latent variable methods: average_variance_extracted(), bifactor_indices(), cfa_2(), cfa_k(), ci_eigenvalue(), common_method_marker(), common_method_single_factor(), dmacs(), ecvi(), htmt(), irt_grm(), irt_information(), measurement_alignment(), measurement_invariance(), procrustes_phi(), simple_structure()

Examples

```
set.seed(113)
f <- rnorm(200)
loadings <- c(0.5, 0.6, 0.65, 0.7, 0.8)
X <- sapply(loadings, function(l) l * f + rnorm(200, sd = sqrt(1 - l^2)))
colnames(X) <- paste0("y", 1:5)

# All columns are items, so the data are all that is needed.
cfa_1(X)

# Equal loadings (essentially tau-equivalent), named in the header.
cfa_1(X, equal_loading = TRUE)

# From a covariance matrix and sample size, as a paper reports them.
cfa_1(S = cov(X), N = 200)

# A subset of columns via \'items\'.
cfa_1(X, items = c("y1", "y2", "y3", "y4"))
```

---

cfa_2 *Two Factor Confirmatory Factor Analysis Model*

Description

Fits a two factor confirmatory factor analysis model to raw item data or a sample covariance matrix. This is the two factor special case of cfa_k: the function is a convenience wrapper that only requires the items of each factor, builds the two factor specification, and forwards everything else to `cfa_k()`. The factors are named `f1` and `f2`, so the rows of the returned table are `lambda_f1_1`, `lambda_f2_1`, `phi_f1_f2` (the factor correlation), `omega_f1`, `omega_f2`, and so on, exactly as a two factor `cfa_k()` call would report them (the `syntax` column names the item behind each number). To name the factors substantively, or for three or more factors, call cfa_k directly.

Usage

```
cfa_2(
  data = NULL,
  factor_1,
  factor_2,
  S = NULL,
  N = NULL,
  equal_loading = FALSE,
  equal_error = FALSE,
  correlated_factors = TRUE,
  estimator = "ML",
  missing = "listwise",
  se = "standard",
  conf_level = 0.95,
```

```
  output = c("verbose", "measurement", "summary", "standardized", "fit"),
  ...
)
```

Arguments

data	A raw data matrix or data frame, rows are respondents and columns include the items named in factor_1 and factor_2. Supply exactly one of data or S.
factor_1	Character vector naming the items of the first factor (two or more).
factor_2	Character vector naming the items of the second factor (two or more). No item may appear in both factors.
S	A symmetric covariance matrix of the items, with dimnames naming the items; N is then required. Supply exactly one of data or S.
N	Total sample size. Required with S; ignored (inferred from the rows) with data.
equal_loading	Logical, or a named logical vector with one element per factor. TRUE constrains the loadings within a factor to a single value (across factors nothing is equated). Defaults to FALSE.
equal_error	Logical, or a named logical vector with one element per factor. TRUE constrains the error variances within a factor to a single value. Defaults to FALSE.
correlated_factors	Logical. If TRUE (default) the factors covary freely; because each factor variance is fixed to 1, the phi terms for factor pairs are the latent correlations. If FALSE the factor covariances are fixed to zero.
estimator	Character; estimator passed to **lavaan**. Must be one of "ML" (default; maximum likelihood, fully efficient under multivariate normality), "MLR" (robust maximum likelihood: maximum likelihood estimates with standard errors and test statistic corrected for nonnormality; Satorra & Bentler, 1994), "WLS" (the asymptotic distribution free estimator of Browne, 1984; raw data and a large sample required), "WLSMV" (diagonally weighted least squares with mean- and variance-adjusted test statistic; Muthén, 1984; Muthén, du Toit, & Spisic, 1997; the standard choice for ordered categorical items, and what ordered switches to), or "GLS" (generalized least squares; Browne, 1974). With a robust estimator the reported fit indices are the robust versions.
missing	Character; missing-data handling passed to **lavaan** when raw data are supplied. Common values are "listwise" (default, listwise deletion) and "ml" (full information maximum likelihood). Ignored with S. With ordered items, "ml"/"fiml" are not available; use "pairwise" or "listwise".
se	Standard error type passed to **lavaan**; see cfa. Common values are "standard" (default), "robust.sem" (with estimator = "MLR"), and "none" (point estimates only; fastest).
conf_level	Confidence level for the parameter confidence intervals, including the delta method intervals for omega, AVE, and *H*. Defaults to 0.95. The RMSEA interval is a separate convention (see Details).
output	Format of the returned object:

"verbose" (default) Parameter estimates with confidence intervals, the per-factor defined parameters (loading_sum, error_sum, omega, ave, H), and fit information.

"measurement" The measurement-property rows only: per factor omega, ave, and H (with delta method standard errors and confidence intervals), the latent correlation phi for every factor pair (with its confidence interval), and, for raw data, the heterotrait-monotrait ratio htmt for every factor pair via htmt.

"summary" The raw summary() output from **lavaan** (not a data frame).

"standardized" The standardized parameter estimates from lavaan::standardizedSolution().

"fit" The raw **lavaan** fit object. Escape hatch for direct lavaan access, including likelihood ratio tests between two cfa_k() fits via lavaan::lavTestLRT().

... Additional arguments forwarded to cfa_k and, through it, to lavaan (for example ordered, equal_intercept, or M).

Details

Each factor is identified by fixing its variance to 1 and estimating every loading; each item loads on exactly one factor (simple structure). The factor correlation is estimated by default and correlated_factors = FALSE fixes it to zero. Per-factor constraint vectors use the factor names, for example equal_loading = c(f1 = TRUE, f2 = FALSE).

Value

The value of the corresponding cfa_k call: a data.frame (classes dmar_cfa_k, dmar_tbl) with one row per parameter (estimate, se, z_value, p_value, ci_lower, ci_upper) followed by the fit rows, or the alternative shapes selected by output (see ?cfa_k).

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

cfa_k for the general function this wraps; cfa_1 for the one factor wrapper; htmt and average_variance_extracted for the discriminant and convergent validity summaries the output = "measurement" table reports alongside omega.

Other multivariate and latent variable methods: average_variance_extracted(), bifactor_indices(), cfa_1(), cfa_k(), ci_eigenvalue(), common_method_marker(), common_method_single_factor(), dmacs(), ecvi(), htmt(), irt_grm(), irt_information(), measurement_alignment(), measurement_invariance(), procrustes_phi(), simple_structure()

Examples

```
data(holzinger_swineford)

# Two factors, each named by its items.
cfa_2(holzinger_swineford,
      factor_1 = c("t6_paragraph_comprehension", "t7_sentence",
```

```
                "t9_word_meaning"),
      factor_2 = c("t20_deduction", "t22_problem_reasoning",
                "t23_series_completion"))

# The measurement properties: omega, ave, and H per factor, the
# factor correlation, and the htmt ratio.
cfa_2(holzinger_swineford,
      factor_1 = c("t6_paragraph_comprehension", "t7_sentence",
                "t9_word_meaning"),
      factor_2 = c("t20_deduction", "t22_problem_reasoning",
                "t23_series_completion"),
      output = "measurement")
```

`cfa_k` *Multiple-Factor Confirmatory Factor Analysis Model*

Description

Fits a confirmatory factor analysis model with one or more factors, where each factor is specified by naming its indicator variables and the measurement structure is specified by describing what is constrained (`equal_loading`, `equal_intercept`, `equal_error`) rather than by more technical terms. The function then reports which classical measurement structure the description implies (congeneric, essentially tau-equivalent, tau-equivalent, essentially parallel, or parallel), the parameter estimates with confidence intervals, fit information, and, per factor, coefficient omega, the average variance extracted (AVE), and coefficient *H*, each with a delta method standard error and confidence interval computed by **lavaan** from defined parameters (no additional packages are involved).

Usage

```
cfa_k(
  data = NULL,
  factors,
  S = NULL,
  N = NULL,
  M = NULL,
  equal_loading = FALSE,
  equal_intercept = FALSE,
  equal_error = FALSE,
  correlated_factors = TRUE,
  meanstructure = NULL,
  estimator = "ML",
  missing = "listwise",
  ordered = NULL,
  se = "standard",
  conf_level = 0.95,
  output = c("verbose", "measurement", "summary", "standardized", "fit"),
```

```
  ...
)
```

Arguments

data	A raw data matrix or data frame, rows are respondents and columns include the items named in factors. Supply exactly one of data or S.
factors	Named list. Each element names a factor and gives the character vector of its indicator columns (two or more per factor; three or more when only one factor is specified). Each item loads on exactly one factor (simple structure).
S	A symmetric covariance matrix of the items, with dimnames naming the items; N is then required. Supply exactly one of data or S.
N	Total sample size. Required with S; ignored (inferred from the rows) with data.
M	Optional named numeric vector of item means, used with a covariance matrix to model the mean structure (required there when equal_intercept is used). Ignored for raw data.
equal_loading	Logical, or a named logical vector with one element per factor. TRUE constrains the loadings within a factor to a single value (across factors nothing is equated). Defaults to FALSE.
equal_intercept	Logical, or a named logical vector with one element per factor. TRUE constrains the item intercepts within a factor to a single value; this requires the mean structure (raw data, or M with a covariance matrix). Defaults to FALSE.
equal_error	Logical, or a named logical vector with one element per factor. TRUE constrains the error variances within a factor to a single value. Defaults to FALSE.
correlated_factors	Logical. If TRUE (default) the factors covary freely; because each factor variance is fixed to 1, the phi terms for factor pairs are the latent correlations. If FALSE the factor covariances are fixed to zero.
meanstructure	Logical or NULL. NULL (default) models the mean structure exactly when it is needed: when any equal_intercept is TRUE or when M is supplied. Set TRUE to model intercepts regardless (raw data or M required), or FALSE to suppress them (an error if equal_intercept is used).
estimator	Character; estimator passed to **lavaan**. Must be one of "ML" (default; maximum likelihood, fully efficient under multivariate normality), "MLR" (robust maximum likelihood: maximum likelihood estimates with standard errors and test statistic corrected for nonnormality; Satorra & Bentler, 1994), "WLS" (the asymptotic distribution free estimator of Browne, 1984; raw data and a large sample required), "WLSMV" (diagonally weighted least squares with mean- and variance-adjusted test statistic; Muthén, 1984; Muthén, du Toit, & Spisic, 1997; the standard choice for ordered categorical items, and what ordered switches to), or "GLS" (generalized least squares; Browne, 1974). With a robust estimator the reported fit indices are the robust versions.
missing	Character; missing-data handling passed to **lavaan** when raw data are supplied. Common values are "listwise" (default, listwise deletion) and "ml" (full information maximum likelihood). Ignored with S. With ordered items, "ml"/"fiml" are not available; use "pairwise" or "listwise".

`ordered` Ordered categorical items: `NULL` (none, the default), `TRUE` (every item), or a character vector of item names. Requires raw data. Each factor must be all ordered or all continuous. Declaring ordered items switches the estimator to `"WLSMV"` (with a message, unless a categorical estimator was requested), fits thresholds in place of intercepts, and reports each ordered factor's omega on the categorical sum score metric via the Green and Yang (2009) computation; see *Details*.

`se` Standard error type passed to **lavaan**; see `cfa`. Common values are `"standard"` (default), `"robust.sem"` (with `estimator = "MLR"`), and `"none"` (point estimates only; fastest).

`conf_level` Confidence level for the parameter confidence intervals, including the delta method intervals for omega, AVE, and *H*. Defaults to 0.95. The RMSEA interval is a separate convention (see Details).

`output` Format of the returned object:

`"verbose"` (default) Parameter estimates with confidence intervals, the per-factor defined parameters (`loading_sum`, `error_sum`, `omega`, `ave`, `H`), and fit information.

`"measurement"` The measurement-property rows only: per factor `omega`, `ave`, and `H` (with delta method standard errors and confidence intervals), the latent correlation `phi` for every factor pair (with its confidence interval), and, for raw data, the heterotrait-monotrait ratio `htmt` for every factor pair via `htmt`.

`"summary"` The raw `summary()` output from **lavaan** (not a data frame).

`"standardized"` The standardized parameter estimates from `lavaan::standardizedSolution()`.

`"fit"` The raw **lavaan** fit object. Escape hatch for direct lavaan access, including likelihood ratio tests between two `cfa_k()` fits via `lavaan::lavTestLRT()`.

`...` Additional arguments forwarded to `lavaan` (e.g., `group`, `cluster`, `bootstrap`).

Details

With `ordered` items, the model is fit by WLSMV to polychoric correlations with thresholds. A sum score of ordered items lives on the metric of the observed categories, not on the latent response metric of the polychoric loadings, so for a factor whose items are ordered the reported omega is the Green and Yang (2009) categorical sum score omega computed from the same fit; the substitution is announced in a message and recorded in the `omega_metric` attribute, and the delta method interval columns are `NA` for those rows because the delta method interval describes the latent response metric. The categorical sum score omega is the coefficient Kelley and Pornprasertmanit (2016) call categorical omega. AVE and coefficient *H* concern the latent response correlations themselves and are reported unchanged on that metric. For a bootstrap confidence interval on a categorical omega, use `reliability_omega_categorical` on the factor's items.

`cfa_1` and `cfa_2` are convenience wrappers around this function for the one and two factor cases: `cfa_1()` takes a vector of items and fits one factor over them, and `cfa_2()` takes the items of each of two factors. Both forward every argument here, so their results are this function's results, with the factors named `f1` (and `f2`).

Describing the model instead of naming it. The classical measurement structures are nested patterns of within-factor equality constraints (Lord & Novick, 1968; Graham, 2006):

congeneric loadings, intercepts, and error variances all free.

essentially tau-equivalent equal loadings; intercepts and error variances free.

tau-equivalent equal loadings and equal intercepts; error variances free.

essentially parallel equal loadings and equal error variances; intercepts free.

parallel equal loadings, equal intercepts, and equal error variances.

The caller states the constraints; the function reports the implied name, per factor, in the printed header and in the `"model"` attribute of the returned table. The distinction between tau-equivalent and essentially tau-equivalent (and between parallel and essentially parallel) lives entirely in the mean structure: the covariance structure of the two members of each pair is identical, so without intercepts in the model only the "essentially" form can be claimed. That is why `equal_intercept` requires raw data or `M`: covariances alone cannot speak to it. A constraint pattern outside the classical list (for example equal error variances with free loadings) is fit as requested and labeled descriptively, since it has no conventional name.

Identification fixes each factor variance to 1 (and each factor mean to 0 when the mean structure is modeled), so all loadings are estimated and within-factor equality constraints are meaningful. Two `cfa_k()` fits that differ only in descriptor settings are nested, so `output = "fit"` feeds `lavaan::lavTestLRT()` directly (with `estimator = "MLR"`, lavaan applies the scaled difference test).

Measurement properties. For factor f with unstandardized loadings λ_j and error variances ψ_j (factor variance 1): coefficient omega $= (\sum_j \lambda_j)^2/((\sum_j \lambda_j)^2 + \sum_j \psi_j)$ (McDonald, 1999), the reliability of the unit-weighted composite; the average variance extracted $= J^{-1} \sum_j \lambda_j^2/(\lambda_j^2 + \psi_j)$ (Fornell & Larcker, 1981), the mean proportion of item variance the factor accounts for; and coefficient $H = (1 + (\sum_j \lambda_j^2/\psi_j)^{-1})^{-1}$ (Hancock & Mueller, 2001), the reliability of the optimally weighted composite, which no single item can drag below its value for any subset. All three are computed as **lavaan** defined parameters, so each carries a delta method standard error and a `conf_level` confidence interval in the same table as the model parameters.

Discriminant validity. Three complementary readings come from `output = "measurement"`: (a) the latent correlation `phi` for a factor pair, with a confidence interval whose upper limit near 1 means the data cannot distinguish the two factors; (b) the Fornell and Larcker (1981) comparison, which asks whether each factor's `ave` exceeds the squared `phi` of its pairs (the factor should share more variance with its own items than with the other factor); and (c) for raw data, the model-free `htmt` ratio (Henseler, Ringle, & Sarstedt, 2015). The rows report the numbers and their uncertainty; the judgment is the researcher's.

Confidence interval conventions. Parameter rows (including omega, `ave`, *H*, and `phi`) use `conf_level`. The RMSEA interval follows its own convention: the `rmsea_ci_level` row records the level actually used (0.90, the conventional level for RMSEA, as in **lavaan**), and `rmsea_ci_lower` / `rmsea_ci_upper` are that interval.

Common row names under `term`: `lambda_<factor>_<j>` (loadings; `lambda_<factor>` when equated), `psi_<factor>_<j>` (error variances; `psi_<factor>` when equated), `nu_<factor>_<j>` (intercepts, when the mean structure is modeled; `nu_<factor>` when equated), `phi_<factor>` (factor variance, fixed to 1), `phi_<factor1>_<factor2>` (latent correlation), the per-factor defined parameters (`loading_sum_<factor>`, `error_sum_<factor>`, `omega_<factor>`, `ave_<factor>`, `H_<factor>`), and the fit rows `chi_square`, `df`, `p_chi_square`, `cfi`, `tli`, `nnfi`, `rmsea`, `rmsea_ci_lower`, `rmsea_ci_upper`, `rmsea_ci_level`, `srmr`, `AIC`, `BIC`, `H0`, `H1`.

Value

For `output = "verbose"` (default) and `output = "measurement"`, a `data.frame` (classes `dmar_cfa_k`, `dmar_tbl`) with columns `syntax`, `term`, `estimate`, `se`, `z_value`, `p_value`, `ci_lower`, `ci_upper`. The `"model"` attribute is a named character vector giving, per factor, the implied classical structure; the printed header displays it. For `output = "summary"`, the **lavaan** summary object; for `"standardized"`, the standardized solution; for `"fit"`, the **lavaan** fit object.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Browne, M. W. (1974). Generalized least squares estimators in the analysis of covariance structures. *South African Statistical Journal, 8*, 1–24.

Browne, M. W. (1984). Asymptotically distribution-free methods for the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology, 37*, 62–83.

Muthén, B. (1984). A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators. *Psychometrika, 49*(1), 115–132.

Muthén, B., du Toit, S. H. C., & Spisic, D. (1997). *Robust inference using weighted least squares and quadratic estimating equations in latent variable modeling with categorical and continuous outcomes*. Unpublished technical report.

Satorra, A., & Bentler, P. M. (1994). Corrections to test statistics and standard errors in covariance structure analysis. In A. von Eye & C. C. Clogg (Eds.), *Latent variables analysis: Applications for developmental research* (pp. 399–419). Thousand Oaks, CA: Sage.

Fornell, C., & Larcker, D. F. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research, 18*(1), 39–50.

Graham, J. M. (2006). Congeneric and (essentially) tau-equivalent estimates of score reliability: What they are and how to use them. *Educational and Psychological Measurement, 66*(6), 930–944. doi:10.1177/0013164406288165

Green, S. B., & Yang, Y. (2009). Reliability of summed item scores using structural equation modeling: An alternative to coefficient alpha. *Psychometrika, 74*(1), 155–167. doi:10.1007/s11336008-90993

Hancock, G. R., & Mueller, R. O. (2001). Rethinking construct reliability within latent variable systems. In R. Cudeck, S. du Toit, & D. Sörbom (Eds.), *Structural equation modeling: Present and future* (pp. 195–216). Scientific Software International.

Henseler, J., Ringle, C. M., & Sarstedt, M. (2015). A new criterion for assessing discriminant validity in variance-based structural equation modeling. *Journal of the Academy of Marketing Science, 43*(1), 115–135. doi:10.1007/s1174701404038

Kelley, K., & Pornprasertmanit, S. (2016). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for composite measures. *Psychological Methods, 21*, 69–92. doi:10.1037/a0040086

Lord, F. M., & Novick, M. R. (1968). *Statistical theories of mental test scores*. Addison-Wesley.

McDonald, R. P. (1999). *Test theory: A unified treatment*. Erlbaum.

See Also

cfa_1 and cfa_2 for the one and two factor convenience wrappers; plot_cfa_k to display the estimates and the equality question visually; reliability_omega, reliability_H, average_variance_extracted, and htmt for the measurement properties as standalone functions; measurement_invariance for the across-group analog of these within-factor constraints; lavaan, lavTestLRT.

Other multivariate and latent variable methods: average_variance_extracted(), bifactor_indices(), cfa_1(), cfa_2(), ci_eigenvalue(), common_method_marker(), common_method_single_factor(), dmacs(), ecvi(), htmt(), irt_grm(), irt_information(), measurement_alignment(), measurement_invariance(), procrustes_phi(), simple_structure()

Examples

```
data(holzinger_swineford)
hs_factors <- list(
  verbal    = c("t6_paragraph_comprehension", "t7_sentence",
                "t9_word_meaning"),
  deduction = c("t20_deduction", "t22_problem_reasoning",
                "t23_series_completion"))

# Congeneric measurement model for both factors (the default:
# nothing is constrained, and the header names the structure).
cfa_k(holzinger_swineford, hs_factors)

# Equal loadings within every factor. Because only the covariance
# structure identifies this constraint, the implied structure is
# essentially tau-equivalent, and the header says so.
cfa_k(holzinger_swineford, hs_factors, equal_loading = TRUE)

# Measurement properties: omega, ave, and H per factor (each with a
# delta method standard error and confidence interval), the latent
# correlations, and the htmt ratios.
cfa_k(holzinger_swineford, hs_factors, output = "measurement")

# Descriptors can differ by factor: here the verbal loadings are
# equated and the deduction loadings are left free, and the header
# names each factor's structure separately.
cfa_k(holzinger_swineford, hs_factors,
      equal_loading = c(verbal = TRUE, deduction = FALSE))

# Equal loadings and intercepts (tau-equivalent), then also equal
# error variances (parallel). The mean structure is added because
# equal_intercept asks about it, so the nu terms join the table.
cfa_k(holzinger_swineford, hs_factors, equal_loading = TRUE,
      equal_intercept = TRUE)
cfa_k(holzinger_swineford, hs_factors, equal_loading = TRUE,
      equal_intercept = TRUE, equal_error = TRUE)

# Ordered categorical items: two correlated factors of three
# four-category items each, generated here. The model is fit by WLSMV
# to polychoric correlations, and each ordered factor's omega is
# reported on the categorical sum score metric (Green & Yang, 2009),
```

```
# which is why the interval columns of those two rows are NA.
set.seed(113)
n <- 200
eta_a <- rnorm(n)
eta_b <- 0.5 * eta_a + sqrt(1 - 0.5^2) * rnorm(n)
lambda <- c(0.6, 0.7, 0.8)
lat <- cbind(outer(eta_a, lambda), outer(eta_b, lambda)) +
  matrix(rnorm(n * 6), n, 6) %*% diag(sqrt(1 - c(lambda, lambda)^2))
likert <- as.data.frame(apply(lat, 2, function(x)
  as.integer(cut(x, breaks = c(-Inf, -1, 0, 1, Inf)))))
names(likert) <- paste0("item_", 1:6)
cfa_k(likert,
      list(scale_a = paste0("item_", 1:3),
           scale_b = paste0("item_", 4:6)),
      ordered = TRUE, output = "measurement")

# Does the equal-loadings description hold? Two fits that differ only
# in a descriptor are nested, so output = "fit" hands them straight to
# lavaan's likelihood ratio test.
fit_free  <- cfa_k(holzinger_swineford, hs_factors, output = "fit")
fit_equal <- cfa_k(holzinger_swineford, hs_factors,
                   equal_loading = TRUE, output = "fit")
lavaan::lavTestLRT(fit_free, fit_equal)
```

`ci_c` *Confidence Interval for a Contrast in a Fixed Effects ANOVA*

Description

Computes the confidence interval for an unstandardized contrast of means in a fixed effects analysis of variance, so a focused comparison among groups (a pairwise difference or any weighted combination of the means) is reported with its precision and in the units of the response. Homogeneity of variance is assumed, as in the ANOVA on which `s_anova` is based.

Usage

```
ci_c(
  means = NULL,
  s_anova = NULL,
  c_weights = NULL,
  n = NULL,
  N = NULL,
  psi = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  df_error = NULL,
  ...
)
```

Arguments

means	A vector of the group means or the means of the particular level of the effect (for fixed effect designs)
s_anova	The standard deviation of the errors from the ANOVA model (i.e., the square root of the mean square error)
c_weights	The contrast weights (choose weights so that the positive *c*-weights sum to 1 and the negative *c*-weights sum to -1; i.e., use fractional values not integers)
n	Sample sizes *per group* or level of the particular factor (if length 1 it is assumed that the per group/level sample sizes are equal)
N	Total sample size
psi	The contrast effect, obtained by multiplying the *j*th mean by the *j*th contrast weight.
conf_level	Confidence interval coverage (i.e., 1- Type I error rate); default is .95
alpha_lower	Type I error for the lower confidence limit
alpha_upper	Type I error for the upper confidence limit
df_error	The degrees of freedom for the error. In one-way designs, this is simply *N*-length (means) and need not be specified; it must be specified if the design has multiple factors.
...	Allows one to potentially include parameter values for inner functions

Value

A 3-row data.frame with columns term and value. The term values are "lower_limit" (the lower confidence limit on the population contrast), "contrast" (the estimated unstandardized contrast), and "upper_limit" (the upper limit).

Note

Be sure to use the standard deviation and not the error variance for s_anova, not the square of this value (the error variance) which would come from the source table (i.e., use the root mean square error, not the mean square error).

Be sure to use fractional *c*-weights when doing complex contrasts (not integers) to specify c_weights. For example, in an ANCOVA of four groups, if the user wants to compare the mean of group 1 and 2 with the mean of group 3 and 4, c_weights should be specified as c(0.5, 0.5, -0.5, -0.5) rather than c(1, 1, -1, -1). Make sure the sum of the contrast weights is zero.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means.)

Steiger, J. H. (2004). Beyond the *F* Test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

ci_sc

Other confidence intervals for effect sizes: ci_R2(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
# Here is a four group example. Suppose that the means of groups 1--4 are 2, 4, 9,
# and 13, respectively. Further, let the error variance be .64 and thus the standard
# deviation would be .80 (note we use the standard deviation in the function, not the
# variance). The contrast of interest here is the average of groups 1 and 4 versus the
# average of groups 2 and 3.
ci_c(means = c(2, 4, 9, 13), s_anova = .80, c_weights = c(.5, -.5, -.5, .5),
n = c(3, 3, 3, 3), N = 12, conf_level = .95)

# Here is an example with two groups.
ci_c(means = c(1.6, 0), s_anova = .80, c_weights = c(1, -1),
n = c(10, 10), N = 20, conf_level = .95)

# An example given by Maxwell, Delaney, & Kelley (2027) :
# 20 subjects of mild hypertensives are assigned to one of four treatments: drug
# therapy, biofeedback, dietary modification, and a treatment combining all the
# three previous treatments. Subjects' blood pressure is measured two weeks
# after the termination of treatment. Now we want to form a 95% level
# confidence interval for the difference in blood pressure between subjects
# who received drug treatment and those who received biofeedback treatment

## Drug group's mean = 94; group size=4
## Biofeedback group's mean = 91; group size=6
## Diet group's mean = 92; group size=5
## Combination group's mean = 83; group size=5
## Mean Square Within (i.e., 'error_variance') = 67.375

ci_c(means = c(94, 91, 92, 83), s_anova = sqrt(67.375), c_weights = c(1, -1, 0, 0),
n = c(4, 6, 5, 5), N = 20, conf_level = .95)
```

ci_c_ancova *Confidence Interval for an (Unstandardized) Contrast in ANCOVA With One Covariate*

Description

Calculates the confidence interval for an unstandardized contrast in the one-covariate ANCOVA. Two procedures are available through the `procedure` argument. The default (`"t"`) returns a single *per-comparison* interval based on the *t* distribution: it gives the correct $(1-$ `conf_level`$)$ coverage for one contrast chosen in advance, and its standard error includes the $(\sum c_i \bar{X}_i)^2 / SS_{\text{within}(x)}$ term that accounts for the covariate separation between the groups in that one contrast. The `"bryant_paulson"` procedure instead returns Bryant–Paulson *simultaneous* (familywise) intervals over a whole family of contrasts of adjusted means; when selected, `ci_c_ancova` simply forwards its arguments to `ci_c_ancova_bp` and returns that result. See Details for which to use when.

Usage

```
ci_c_ancova(
  psi = NULL,
  adj_means = NULL,
  s_ancova = NULL,
  c_weights,
  n,
  cov_means,
  SSwithin_x,
  conf_level = 0.95,
  procedure = c("t", "bryant_paulson"),
  ...
)
```

Arguments

`psi`	The unstandardized contrast of adjusted means
`adj_means`	The vector that contains the adjusted mean of each group on the dependent variable
`s_ancova`	The standard deviation of the errors from the ANCOVA model (i.e., the square root of the mean square error from ANCOVA)
`c_weights`	The contrast weights
`n`	Either a single number that indicates the sample size *per group* or a vector that contains the sample size of each group
`cov_means`	A vector that contains the group means of the covariate
`SSwithin_x`	The sum of squares within groups obtained from the summary table for ANOVA on the covariate
`conf_level`	The desired confidence interval coverage, (i.e., 1 - Type I error rate)

procedure	The interval procedure, one of "t" (the default, a single per-comparison interval based on the *t* distribution) or "bryant_paulson" (Bryant–Paulson simultaneous intervals for a family of contrasts of adjusted means). When "bryant_paulson" is chosen the call is forwarded to ci_c_ancova_bp; see Details.
...	Allows one to potentially include parameter values for inner functions. When procedure = "bryant_paulson", these are passed on to ci_c_ancova_bp (for example num_covariates, df, or contrast_type).

Details

Per-comparison versus simultaneous. The two procedures answer different questions and are not interchangeable. Use the default procedure = "t" when a single contrast was planned in advance: the interval has exact per-comparison coverage and its width reflects the covariate adjustment for that specific contrast through the $(\sum c_i \bar{X}_i)^2 / SS_{\text{within}(x)}$ term. Use procedure = "bryant_paulson" when several contrasts (for example all pairwise comparisons of adjusted means) are examined together and the coverage statement must hold simultaneously across the family: the Bryant–Paulson generalized studentized range supplies a larger critical value that controls the familywise error rate and, because the covariates are random, correctly absorbs the extra sampling uncertainty from estimating the covariate adjustment (which holds on average over the covariate distribution, so the per-contrast separation term is not added again). A per-comparison interval used as if it were simultaneous understates the family error rate; a simultaneous interval used for one planned contrast is wider than necessary. See ci_c_ancova_bp for the full description of the simultaneous procedure and its arguments.

Value

A 3-row data.frame with columns term and value (numeric). The term values are "lower_limit" (the lower confidence limit on the unstandardized ANCOVA contrast), "psi" (the unstandardized contrast point estimate), and "upper_limit" (the upper limit).

When procedure = "bryant_paulson", the return value is whatever ci_c_ancova_bp returns (a table with one row per contrast and columns contrast, estimate, lower_limit, and upper_limit).

Note

Be sure to use the standard deviation and not the error variance for s_ancova, not the square of this value which would come from the source table (i.e., do not use the variance of the error but rather use the square root).

If n receives a single number, that number is considered as the sample size *per group*. If n receives a vector, the vector is considered as the sample size of each group.

Be sure to use fractions not the integers to specify c_weights. For example, in an ANCOVA of four groups, if the user wants to compare the mean of group 1 and 2 with the mean of group 3 and 4, c_weights should be specified as c(0.5, 0.5, -0.5, -0.5) rather than c(1, 1, -1, -1). Make sure the sum of the contrast weights are zero.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9.)

See Also

ci_c,ci_sc_ancova, ci_c_ancova_bp for the Bryant–Paulson simultaneous procedure

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
# Maxwell, Delaney, & Kelley (2027) offer an example that 30 depressive
# individuals are randomly assigned to three groups, 10 in each, and ANCOVA
# is performed on the posttest scores using the participants' pretest
# scores as the covariate. The means of pretest scores of group 1 to 3 are
# 17, 17.7, and 17.4, respectively, and the adjusted means of groups 1 to 3
# are 7.5, 12, and 14, respectively. The error variance in ANCOVA is 29,
# and the sum of squares within groups from ANOVA on the covariate is 752.5.

# To obtain the confidence interval for adjusted mean of group 1 versus group 2:
ci_c_ancova(adj_means = c(7.5, 12, 14), s_ancova = sqrt(29),
            c_weights = c(1, -1, 0), n = 10,
            cov_means = c(17, 17.7, 17.4), SSwithin_x = 752.5)

# That interval is the right one for a single contrast planned in advance.
# For the family of all three pairwise comparisons of the adjusted means,
# with coverage that holds simultaneously across the family, select
# procedure = "bryant_paulson"; the call is forwarded to ci_c_ancova_bp().
# The simultaneous limits are wider, which is the price of the family
# statement: group 1 against group 2 runs from -10.61 to 1.61, where the
# single planned contrast above ran from -9.46 to 0.46.
ci_c_ancova(adj_means = c(7.5, 12, 14), s_ancova = sqrt(29), n = 10,
            procedure = "bryant_paulson")
```

`ci_c_ancova_bp` *Bryant–Paulson Simultaneous Confidence Intervals for Contrasts of Adjusted Means in ANCOVA*

Description

Constructs Tukey–Kramer-type *simultaneous* confidence intervals on one or more contrasts of covariate-adjusted means in the analysis of covariance (ANCOVA) when the covariate(s) are *random*, using the Bryant–Paulson generalized studentized range (see `bryant_paulson`). Unlike the per-comparison interval of `ci_c_ancova`, these intervals control the *familywise* error rate over the whole family of comparisons and, through the Bryant–Paulson critical value, correctly account for the extra sampling uncertainty that comes from estimating the covariate adjustment from random covariates. Naively applying Tukey's method to adjusted means ignores that uncertainty and produces intervals that are too narrow (below-nominal coverage); a simulation study of that undercoverage is maintained alongside the package.

Usage

```
ci_c_ancova_bp(
  adj_means,
  s_ancova,
  c_weights = NULL,
  n,
  num_covariates = 1,
  df = NULL,
  conf_level = 0.95,
  contrast_type = c("pairwise", "allowance"),
  ...
)
```

Arguments

`adj_means`	A numeric vector of the covariate-adjusted group means (one per group). The number of groups k is inferred from its length.
`s_ancova`	The standard deviation of the errors from the ANCOVA model, i.e., the square root of the ANCOVA error mean square (use the standard deviation, *not* the variance from the source table).
`c_weights`	Optional contrast weights. May be (i) a numeric vector of length k giving a single contrast, or (ii) a matrix/data.frame with k columns, each row a contrast. If `NULL` (the default), all $k(k-1)/2$ pairwise comparisons are returned. For each contrast the weights should sum to zero.
`n`	Either a single number giving the common per-group sample size or a numeric vector of per-group sample sizes. The Bryant–Paulson distribution is exact for balanced designs; for unequal n a Tukey–Kramer harmonic adjustment is used (see Details).
`num_covariates`	The number of random covariates, p. Default `1`.

df	Optional error degrees of freedom ν. If NULL (default) it is computed for a one-way ANCOVA as $\nu = \sum n - k - p$. Supply it directly for other designs (e.g., a randomized-block ANCOVA, where ν differs).
conf_level	The simultaneous (familywise) confidence level. Default 0.95.
contrast_type	One of "pairwise" (default) or "allowance". "pairwise" uses the Tukey–Kramer quadratic standard error, exact for pairwise comparisons. "allowance" uses Tukey's allowance, which yields intervals that hold simultaneously over *all* contrasts, including complex ones (this is the form in Eq. (2.4) of Bryant and Bruvold, 1980). The width factor each choice applies is given in Details. The two coincide for pairwise comparisons.
...	Additional arguments (currently unused).

Details

The interval. For a contrast $\psi = \sum_i c_i \theta_i$ of adjusted means, the simultaneous interval is

$$\hat{\psi} \;\pm\; q_{\alpha;\,p,k,\nu}\;\hat{\sigma}_{y|x}\;w(c),$$

where $q_{\alpha;\,p,k,\nu}$ is the upper-α Bryant–Paulson critical value (qbryant_paulson) and the width factor is $w(c) = \frac{1}{\sqrt{2}}\sqrt{\sum_i c_i^2/n_i}$ for contrast_type = "pairwise" or $w(c) = \frac{1}{2}\sum_i |c_i|\sqrt{1/n}$ for contrast_type = "allowance" (balanced n). For a pairwise difference with common n both reduce to $q_{\alpha;\,p,k,\nu}\,\hat{\sigma}_{y|x}\sqrt{1/n}$, reproducing the critical difference of Bryant and Bruvold (1980).

No per-comparison covariate term. By design the standard error here does *not* include the $(\bar{X}_i - \bar{X}_j)^2/SS_{\text{within}(x)}$ term that appears in a single-comparison ANCOVA interval (ci_c_ancova). In the Bryant–Paulson framework the random-covariate uncertainty is carried by the (larger) critical value, which holds *on average* over the covariate distribution; adding the per-pair term as well would double-count it.

Unequal sample sizes. For unbalanced designs the "pairwise" standard error uses $\sqrt{c_i^2/n_i}$ directly (the Tukey–Kramer generalization); coverage is then approximate but typically very close to nominal and slightly conservative.

Value

A data.frame (class dmar_tbl) with one row per contrast and columns contrast (a label), estimate (the contrast of adjusted means $\hat{\psi}$), lower_limit, and upper_limit. The Bryant–Paulson critical value used is stored in the "critical_value" attribute and the confidence level in the "conf_level" attribute (printed beneath the table).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bryant, J. L., & Paulson, A. S. (1976). An extension of Tukey's method of multiple comparisons to experimental designs with random concomitant variables. *Biometrika, 63*, 631–638.

Bryant, J. L., & Bruvold, N. T. (1980). Multiple comparison procedures in the analysis of covariance. *Journal of the American Statistical Association, 75*(372), 874–880. doi:10.2307/2287175

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9.)

See Also

bryant_paulson for the underlying distribution; ci_c_ancova for the per-comparison interval; ancova for an ANCOVA fit.

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
# Bryant and Bruvold (1980) worked example: 6 panels, 1 covariate, nu = 14,
# ANCOVA error MS = 0.01326. Here the design is a randomized block with
# s = 4 blocks, so the per-group "n" for the adjusted-mean SE is 4 and the
# error df (14) must be supplied directly. The multiplier for these
# intervals is a Bryant-Paulson quantile, which has no closed form; it is
# obtained by inverting the distribution function with a root search.
adj <- c(3.595, 3.619, 4.102, 4.515, 4.618, 4.876)
bp <- ci_c_ancova_bp(adj_means = adj, s_ancova = sqrt(0.01326),
                     n = 4, num_covariates = 1, df = 14)
bp
# The multiplier is 4.83 and every pairwise critical difference is 0.278,
# matching the paper; the 15 intervals hold jointly at the 95 percent level.
# The multiplier is kept on the result, so the critical difference can be
# rebuilt by hand as q * s_ancova * sqrt(1/n):
attr(bp, "critical_value")
attr(bp, "critical_value") * sqrt(0.01326) * sqrt(1 / 4)

# A complex contrast (say panels 1 and 2 against panels 3 through 6) is
# requested by passing its weights to c_weights, together with
# contrast_type = "allowance", the all-contrasts form of Eq. (2.4) of
# Bryant and Bruvold. That contrast of adjusted means is -0.921, with
# simultaneous limits of -1.199 and -0.643.
ci_c_ancova_bp(adj_means = adj, s_ancova = sqrt(0.01326),
               c_weights = c(0.5, 0.5, -0.25, -0.25, -0.25, -0.25),
               n = 4, num_covariates = 1, df = 14,
               contrast_type = "allowance")
```

---

ci_correlation *Confidence Intervals for the Population Correlation and Multiple Correlation*

Description

Two confidence intervals on the correlation scale share this page, named by the convention that lowercase *r* is the Pearson product-moment correlation between two variables and capital *R* is the multiple correlation between an outcome and a set of predictors.

`ci_r()` forms a confidence interval for the population correlation coefficient ρ. The confidence interval is for the population value ρ; the required input is the corresponding sample value, the observed sample correlation coefficient *r*. This approach assumes that the two variables on which the correlation is based are bivariate normally distributed (e.g., Hays, 1994, Chapter 14).

`ci_R()` constructs a confidence interval for the population multiple correlation coefficient $\rho = \sqrt{\rho^2}$ from the sample multiple correlation coefficient (or, equivalently, from the observed F-statistic and degrees of freedom). The interval is obtained by inverting the sampling distribution of the sample R^2 and propagating the limits through the monotone (square root) transform.

The two estimands meet at a single predictor: the multiple correlation from a regression on one predictor is the absolute value of the Pearson correlation between the outcome and that predictor.

Usage

```
ci_r(r, n, conf_level = 0.95, alpha_lower = NULL, alpha_upper = NULL)

ci_R(
  R = NULL,
  df_1 = NULL,
  df_2 = NULL,
  conf_level = 0.95,
  random_predictors = TRUE,
  F_value = NULL,
  N = NULL,
  p = NULL,
  alpha_lower = NULL,
  alpha_upper = NULL,
  ...
)
```

Arguments

`r`	Observed value of the sample correlation coefficient (specifically the zero-order Pearson product-moment correlation coefficient), for `ci_r()`
`n`	Sample size for `ci_r()`, which must be at least 4 (see Details)
`conf_level`	Confidence interval coverage (i.e., 1 - Type I error rate); default is .95
`alpha_lower`	The Type I error rate for the lower confidence interval limit
`alpha_upper`	The Type I error rate for the upper confidence interval limit
`R`	Observed value of the sample multiple correlation coefficient, for `ci_R()`
`df_1`	Numerator degrees of freedom
`df_2`	Denominator degrees of freedom

random_predictors	Whether or not the predictor variables are random or fixed (random is default)
F_value	Obtained F-value
N	Sample size
p	Number of predictors
...	Allows one to potentially include parameter values for inner functions

Details

The Pearson correlation interval (ci_r)**.** This approach will not generally lead to a symmetric confidence interval. The function first transforms r into Z', forms a confidence interval for the population value (i.e., ζ), and then transforms the confidence limits for ζ into the scale of the correlation coefficient. The interval requires a sample size of at least 4. The variance of Z' is $1/(n-3)$, which is infinite at $n = 3$; there the interval would be vacuous, covering $[-1, 1]$ regardless of *r*, and for smaller *n* the variance is undefined. The function therefore stops with an error when $n < 4$.

Fixed vs. random predictors (ci_R)**.** The two regression models give *different* sampling distributions for the sample R^2, and so different confidence intervals on ρ. Under fixed predictors the design matrix is treated as constant in hypothetical replications of the study, and the omnibus F-statistic follows a noncentral *F* with p and $N-p-1$ degrees of freedom and noncentrality $\lambda = N\rho^2/(1-\rho^2)$ (Cohen, 1988); the CI on ρ^2 is obtained by inverting that distribution and then taking the square root (see ci_nc_F). Under random predictors the design matrix is itself a draw from a joint multivariate normal distribution and the unconditional sampling distribution of the sample R^2 is given by Lee (1971); the same Lee bisection that ci_R2 uses for the random-predictor CI on ρ^2 is applied here and the limits are mapped to ρ. Gatsonis and Sampson (1989) document the comparison; in the behavioral, educational, and social sciences predictor variables are almost always random, so the default is random_predictors = TRUE. Pass random_predictors = FALSE for designs in which the predictor variables are fixed by design.

Value

ci_r() returns a 3-row data.frame with columns term and value. The term values are "lower_limit" (the lower confidence limit on the population correlation ρ), "r" (the observed sample correlation coefficient), and "upper_limit" (the upper limit on ρ).

ci_R() returns a 3-row data.frame with columns term, value, prob_less, and prob_greater. The rows are ordered "lower_limit", "R" (the sample multiple correlation coefficient supplied by the user, the point estimate), and "upper_limit", so the point estimate sits between its confidence limits. The lower and upper limits are the confidence limits on the population multiple correlation coefficient ρ (square roots of the corresponding limits on ρ^2). The prob_less and prob_greater columns report the achieved lower-tail and upper-tail error probabilities at each limit (they are NA for the "R" estimate row).

Note

The ci_r() confidence interval assumes that the two variables the correlation is based on are bivariate normal. See Hays (1994, Chapter 14) for details.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Algina, J. & Olejnik, S. (2000). Determining sample size for accurate estimation of the squared multiple correlation coefficient. *Multivariate Behavioral Research, 35*, 119–137. doi:10.1207/s15327906mbr3501_5

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Gatsonis, C., & Sampson, A. R. (1989). Multiple correlation: Exact power and sample size calculations. *Psychological Bulletin, 106*(3), 516–524.

Hays, W. L. (1994). *Statistics* (5th ed.). Fort Worth, TX: Harcourt Brace College Publishers.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*, 524–555. doi:10.1080/00273170802490632

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Lee, Y. S. (1971). Some results on the sampling distribution of the multiple correlation coefficient. *Journal of the Royal Statistical Society, Series B, 33*(1), 117–130.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on R^2 as a model comparison effect size.)

Smithson, M. (2003). *Confidence intervals*. Thousand Oaks, CA: Sage Publications.

Steiger, J. H. (2004). Beyond the *F* test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

Steiger, J. H., & Fouladi, R. T. (1992). R2: A computer program for interval estimation, power calculations, sample size estimation, and hypothesis testing in multiple regression. *Behavior Research Methods, Instruments, & Computers, 24*(4), 581–582. doi:10.3758/BF03203611

See Also

ci_R2, ss_aipe_r, ss_power_r, var_r, ss_aipe_R2, convert_r_Z, convert_Z_r, ci_nc_t

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
# Pearson correlation, from Hays. Suppose n = 100 and r = .35.
ci_r(r = .35, n = 100, conf_level = .95)

# Here is another way to enter the above example.
ci_r(r = .35, n = 100, conf_level = NULL,
     alpha_lower = .025, alpha_upper = .025)

# Here are examples of one-sided confidence intervals.
ci_r(r = .35, n = 100, conf_level = NULL, alpha_lower = 0, alpha_upper = .05)
ci_r(r = .35, n = 100, conf_level = NULL, alpha_lower = .05, alpha_upper = 0)

# Multiple correlation from a five-predictor regression.
ci_R(R = .7071, df_1 = 5, df_2 = 50, conf_level = .95,
     random_predictors = TRUE)
```

---

ci_cv *Confidence Interval for the Coefficient of Variation*

Description

Computes the noncentral *t*-based confidence interval for the population coefficient of variation, the standard deviation relative to the mean, so variability can be reported on a scale that is free of the measurement units.

Usage

```
ci_cv(
  cv = NULL,
  mean = NULL,
  sd = NULL,
  n = NULL,
  data = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  ...
)
```

Arguments

cv	Coefficient of variation
mean	Sample mean
sd	Sample standard deviation (square root of the unbiased estimate of the variance
n	Sample size

data	Vector of data for which the confidence interval for the coefficient of variation is to be calculated
conf_level	Desired confidence level (1-Type I error rate)
alpha_lower	The proportion of values beyond the lower limit of the confidence interval (cannot be used with conf_level)
alpha_upper	The proportion of values beyond the upper limit of the confidence interval (cannot be used with conf_level)
...	Allows one to potentially include parameter values for inner functions

Details

Uses the noncentral *t*-distribution to calculate the confidence interval for the population coefficient of variation.

Value

A 4-row data.frame with columns term, value, prob_less, and prob_greater. The rows are ordered so the two point estimates sit between the confidence limits: "lower_limit" (lower confidence limit on the coefficient of variation), "c_of_v" (the sample coefficient of variation), "c_of_v_unbiased" (the unbiased estimator), and "upper_limit" (upper confidence limit). The prob_less and prob_greater columns report the achieved tail probabilities of the noncentral t search at the limit values; they are NA for the point-estimate rows.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Johnson, N. L., & Welch, B. L. (1940). Applications of the non-central *t*-distribution. *Biometrika, 31*(3–4), 362–389. doi:10.1093/biomet/31.34.362

Kelley, K. (2007). Sample size planning for the coefficient of variation from the accuracy in parameter estimation approach. *Behavior Research Methods, 39*(4), 755–766. doi:10.3758/BF03192966

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3.)

McKay, A. T. (1932). Distribution of the coefficient of variation and the extended *t* distribution. *Journal of the Royal Statistical Society, 95*(4), 695–698.

See Also

cv

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
set.seed(113)
N <- 15
X <- rnorm(N, 5, 1)
mean.X <- mean(X)
sd.X <- var(X)^.5

ci_cv(mean = mean.X, sd = sd.X, n = N, alpha_lower = .025,
alpha_upper = .025, conf_level = NULL)
ci_cv(data = X, conf_level = .95)
ci_cv(cv = sd.X / mean.X, n = N, conf_level = .95)
```

---

ci_dunnett *Dunnett's Simultaneous Confidence Intervals Against a Control*

Description

Computes Dunnett's (1955, 1964) simultaneous confidence intervals for the $a - 1$ comparisons of $a - 1$ treatment means against a single control mean, with family-wise coverage at the specified `conf_level`. Returns the result in tidy long form.

Usage

```
ci_dunnett(
  x,
  group = NULL,
  control = NULL,
  alternative = c("two_sided", "less", "greater"),
  conf_level = 0.95
)
```

Arguments

`x`	Either (a) a fitted `lm` or `aov` object with a one-way factor predictor, or (b) a numeric vector of observations, in which case `group` must also be supplied.
`group`	When `x` is a vector, a factor of group labels of the same length.
`control`	Character name of the control level (must be one of the factor levels). If `NULL` (default), the first level is used.
`alternative`	One of `"two_sided"` (default; the base-R spelling `"two.sided"` is accepted as an alias), `"less"`, or `"greater"`.
`conf_level`	Family-wise confidence level. Default `0.95`.

Details

Critical value. The two-sided Dunnett critical value $d^{(2)}_{\alpha,a-1,\nu}$ is obtained from the multivariate *t* distribution with $a-1$ dimensions, common correlation 0.5 (the Dunnett correlation under balanced *n*; the function does not adjust for unequal *n*), and ν error degrees of freedom. The function uses the existing `cv_dunnett()` critical value.

Adjusted *p*-values. Computed exactly from the same equicorrelated multivariate *t* distribution. The one common correlation $1/2$ admits a one-factor representation, so the probability that all comparisons fall inside (or below) the observed statistic collapses to two nested one-dimensional integrals, evaluated by quadrature. The adjusted *p*-value is one minus that probability. The computation is deterministic (no Monte Carlo) and needs no additional package.

Value

A `data.frame` with one row per non-control level. Columns: `contrast`, `mean_difference`, `se`, `t_statistic`, `lower_limit`, `upper_limit`, `p_adjusted`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Dunnett, C. W. (1955). A multiple comparison procedure for comparing several treatments with a control. *Journal of the American Statistical Association, 50*(272), 1096–1121.

Dunnett, C. W. (1964). New tables for multiple comparisons with a control. *Biometrics, 20*(3), 482–491.

Hsu, J. C. (1996). *Multiple comparisons: Theory and methods*. Chapman & Hall.

See Also

`cv_dunnett`, `ci_tukey_kramer`, `ci_scheffe`

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `correlations_test()`, `equivalence_r()`, `equivalence_smd()`, `factorial_anova()`, `manova_split_plot()`, `mauchly_test()`, `mixed_anova()`, `obrien_test()`, `pairwise_within()`, `randomization_test()`, `randomization_test_paired()`, `regions_of_significance()`, `simple_effects_AB()`, `summary_t_test()`, `welch_t()`

Examples

```
# 1. Compare the SSRI and placebo arms of the depression_bdi study
#    against the wait list control:
fit <- lm(bdi_post ~ condition, data = depression_bdi)
ci_dunnett(fit, control = "wait_list")

# 2. One-sided: a treatment that works pulls the posttest BDI down,
#    so the directional alternative is "less":
ci_dunnett(fit, control = "wait_list", alternative = "less")
```

`ci_eigenvalue` *Confidence Interval on the Largest Eigenvalue of a Sample Covariance Matrix*

Description

Computes an asymptotic confidence interval on the largest population eigenvalue λ_1 of a population covariance matrix, given a sample covariance matrix from n observations on p variables under multivariate normality. Useful in principal-components analysis and dimension-reduction settings to gauge whether the largest eigenvalue is well-separated from the second.

Usage

```
ci_eigenvalue(cov_matrix, n = NULL, conf_level = 0.95, k = 1)
```

Arguments

`cov_matrix`	Sample covariance matrix (a symmetric, positive- semidefinite numeric matrix), or a data frame whose columns are the variables (in which case `cov_matrix` is computed internally).
`n`	Sample size (number of rows of the original data). Required when `cov_matrix` is supplied as a matrix; ignored when `cov_matrix` is a data frame (`nrow(cov_matrix)` is used instead).
`conf_level`	Confidence level. Default `0.95`.
`k`	Which eigenvalue (1 = largest, 2 = next, ...) to bracket. Default 1.

Details

Asymptotic distribution. Under multivariate normality with eigenvalues $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_p$, the sample eigenvalues $\hat{\lambda}_j$ are asymptotically independent and approximately normal with mean λ_j and variance $2\lambda_j^2/(n-1)$ when the eigenvalues are simple (well-separated) (Anderson, 2003, Theorem 13.3.1; Muirhead, 1982, Section 9.7). The asymptotic CI is therefore

$$\hat{\lambda}_j \cdot \exp\left(\pm z_{1-\alpha/2}\sqrt{\frac{2}{n-1}}\right),$$

on the multiplicative scale (equivalently, a Wald CI on $\log \lambda_j$ with variance $2/(n-1)$). The log scale is the natural variance-stabilizing transformation for an eigenvalue.

Caveats. The asymptotic CI assumes well-separated population eigenvalues. When the largest two eigenvalues are close, the sample eigenvalue exhibits a "repulsion" phenomenon and the CI is biased (typically too narrow). Diagnostic: if $\hat{\lambda}_1/\hat{\lambda}_2$ is close to 1, the asymptotic CI should not be relied upon; a bootstrap is preferable.

Value

A 3-row `data.frame` with rows ordered `"lower_limit"`, `"eigenvalue"` (the sample eigenvalue point estimate), and `"upper_limit"`, so the point estimate sits between its confidence limits.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Anderson, T. W. (2003). *An introduction to multivariate statistical analysis* (4th ed.). Wiley. (See Chapter 13.)

Muirhead, R. J. (1982). *Aspects of multivariate statistical theory*. Wiley. (See Section 9.7.)

See Also

prcomp, eigen

Other multivariate and latent variable methods: average_variance_extracted(), bifactor_indices(), cfa_1(), cfa_2(), cfa_k(), common_method_marker(), common_method_single_factor(), dmacs(), ecvi(), htmt(), irt_grm(), irt_information(), measurement_alignment(), measurement_invariance(), procrustes_phi(), simple_structure()

Examples

```
# 1. From a data frame:
set.seed(113)
X <- data.frame(matrix(rnorm(200), nrow = 50))
ci_eigenvalue(X, k = 1)

# 2. From an explicit covariance matrix:
S <- cov(X)
ci_eigenvalue(S, n = nrow(X), k = 1)
```

---

ci_eta_squared *Confidence Interval for Eta Squared (Effect Size for ANOVA)*

Description

Computes the point estimate and an exact, noncentrality-based confidence interval for the population eta squared (η^2), the proportion of variance in the dependent variable accounted for by a fixed effect. Accepts either the raw ANOVA summary (*F*, effect df, error df, total *N*) or a fitted model object. Supports both between-subjects designs (aov / lm) and within-subjects / mixed designs (aovlist fits with an Error() term in the formula). For factorial and within-subjects designs the function returns one row per effect with the CI for *partial* η^2 computed against that effect's own error stratum.

Usage

```
ci_eta_squared(
  object = NULL,
  F_value = NULL,
  df_effect = NULL,
  df_error = NULL,
  N = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL
)
```

Arguments

`object`	Optional. A fitted model object of class `aov`, `lm`, or `aovlist` (multi-stratum aov fit, e.g.\ `aov(y ~ A + Error(subject/A), data = d)`). For multi-stratum fits the function walks every error stratum and returns one row per non-`Residuals` effect, identifying the stratum used.
`F_value`	Observed F-value (ignored if `object` is supplied).
`df_effect`	Numerator degrees of freedom for the effect (ignored if `object` is supplied).
`df_error`	Error (residual) degrees of freedom (ignored if `object` is supplied).
`N`	Total sample size, the total number of observations (ignored if `object` is supplied; derived automatically from a fitted model, via `nobs(object)` for single-stratum fits, or, for `aovlist` fits, recovered as one more than the sum of the effect and residual degrees of freedom across every stratum, which is the total number of observations).
`conf_level`	Desired confidence coverage; default `0.95`. Used only when `alpha_lower` and `alpha_upper` are both `NULL`.
`alpha_lower, alpha_upper`	Optional Type I error on the lower and upper side. If both are `NULL`, a symmetric interval at `conf_level` is used. If both are supplied, `conf_level` is recomputed as `1 - alpha_lower - alpha_upper`.

Details

Point estimate. $\hat{\eta}^2 = df_{\text{effect}} \cdot F / (df_{\text{effect}} \cdot F + df_{\text{error}})$, which equals $SS_{\text{effect}} / (SS_{\text{effect}} + SS_{\text{error}})$. In a one-way ANOVA this is also $SS_{\text{effect}} / SS_{\text{total}}$. In a factorial design the same expression gives the per-effect *partial* η^2.

Confidence interval. The CI is constructed by Steiger's (2004) confidence interval transformation principle: a CI for the noncentrality parameter λ of the F distribution is obtained (via `ci_nc_F`) and then mapped through

$$\eta^2_{\text{bound}} = \frac{\lambda_{\text{bound}}}{\lambda_{\text{bound}} + N}.$$

This is the same transformation used by `ci_pvaf` and `ci_omega_squared`; the three functions share CI machinery and differ only in their sample point estimators. When the lower CI on λ is not

identified (i.e., the observed F is below the one-sided critical value), the lower limit on η^2 is set to 0.

Designs supported.

- *Between-subjects ANOVA*: fitted `aov`/`lm`, or raw F/df/N.
- *Within-subjects or mixed ANOVA*: fitted `aovlist`. Each effect's CI is built from its own stratum's F and residual df; `N` is the total number of observations across all strata. The reported `stratum` column identifies which error term each row used.

Sums of squares in factorial designs. When a fitted model is supplied, F-values are read from `anova()` (single-stratum) or `summary()` (multi-stratum), both of which use Type I (sequential) sums of squares in base R. For balanced designs Types I, II, and III agree; for unbalanced designs they differ. If Type II or III F-values are required, compute them with e.g.\ `car::Anova(object, type = 3)` and pass the relevant F, degrees of freedom, and N into the raw-argument interface.

Value

A `data.frame` with one row per effect. Single-stratum fits and the raw interface return columns `effect`, `eta_squared`, `lower_limit`, `upper_limit`, `F_value`, `df_effect`, `df_error`, `N`. `aovlist` (within-subjects / mixed) fits additionally include a `stratum` column. With the raw-argument interface `effect` is `"overall"`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Fleishman, A. I. (1980). Confidence intervals for correlation ratios. *Educational and Psychological Measurement, 40*(3), 659–670.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*, 137–152. doi:10.1037/a0028086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on η^2, Chapter 7 on factorial designs, and Chapter 11 on generalized η^2 for within-subjects designs.)

Smithson, M. (2001). Correct confidence intervals for various regression effect sizes and parameters: The importance of noncentral distributions in computing intervals. *Educational and Psychological Measurement, 61*, 605–632. doi:10.1177/00131640121971392

Steiger, J. H. (2004). Beyond the F test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

eta_squared, ci_eta_squared_partial, ci_omega_squared, ci_pvaf, ci_nc_F

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
# 1. Raw-argument interface. Bargman's (1970) example.
ci_eta_squared(F_value = 11.221, df_effect = 4, df_error = 50, N = 55)

# Same example with a 90% confidence interval.
ci_eta_squared(
  F_value = 11.221, df_effect = 4, df_error = 50, N = 55,
  conf_level = 0.90
)

# 2. One way ANOVA from a fitted model: mean IQ gain differs across
#        the six grades of the pygmalion data (N = 310).
fit_one <- aov(iq_gain ~ factor(grade), data = pygmalion)
ci_eta_squared(fit_one)

# 3. Two-factor ANOVA: partial eta squared per effect for the
#        manipulated expectancy treatment and the measured grade
#        classification (pygmalion data, N = 310). The treatment by
#        grade interaction is weak here (F = 1.19), so the additive
#        model is used.
fit_additive <- aov(iq_8 ~ treatment + factor(grade), data = pygmalion)
ci_eta_squared(fit_additive)

# 4. Within-subjects ANOVA. CI computed against the within-subjects
#        error stratum (the row reports which one via 'stratum').
set.seed(113)
n <- 20
rm_data <- data.frame(
  subject = factor(rep(seq_len(n), each = 3)),
  time    = factor(rep(c("Pre", "Mid", "Post"), n),
                   levels = c("Pre", "Mid", "Post")),
  y       = rnorm(n, sd = 1.5)[rep(seq_len(n), each = 3)] +
            0.7 * rep(1:3, n) + rnorm(n * 3, sd = 1.2)
)
fit_rm <- aov(y ~ time + Error(subject/time), data = rm_data)
ci_eta_squared(fit_rm)
```

ci_eta_squared_generalized *Confidence Interval for Generalized Eta Squared (Approximate)*

Description

Returns the point estimate of generalized eta squared (η^2_G; Olejnik & Algina, 2003) along with an *optional* confidence interval computed by one of two approximate methods. The default is to return only the point estimate (`method = "none"`), because both available CI methods are approximations whose coverage properties have not been broadly validated for this estimand and warrant independent evaluation before being used in substantive inference.

Usage

```
ci_eta_squared_generalized(
  object = NULL,
  observed = NULL,
  SS_effect = NULL,
  SS_observed = NULL,
  SS_error = NULL,
  F_effect = NULL,
  df_effect = NULL,
  F_observed = NULL,
  df_observed = NULL,
  df_error = NULL,
  N = NULL,
  method = c("none", "parametric", "bootstrap"),
  B = 10000L,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  seed = NULL
)
```

Arguments

`object`	Optional. A fitted model object of class `aov`, `lm`, or `aovlist` (multi-stratum aov fit for within-subjects / mixed designs).
`observed`	Character vector of factor names treated as measured.
`SS_effect`, `SS_observed`, `SS_error`	Sums of squares (option 2 in `eta_squared_generalized`).
`F_effect`, `df_effect`, `F_observed`, `df_observed`, `df_error`	*F*-values and degrees of freedom (option 3 in `eta_squared_generalized`).
`N`	Total sample size. Required when `method = "parametric"` and no fitted model is supplied; ignored when `method = "none"` or when a fitted model is supplied (derived automatically, via `nobs(object)` for single-stratum fits, or by summing degrees of freedom across error strata for `aovlist`).

`method`	One of `"none"` (default), `"parametric"`, or `"bootstrap"`. See Details.
`B`	Integer. Number of bootstrap replications when `method = "bootstrap"`. Minimum `1000` (enforced); default `10000`. We recommend `10000` or more for publication-quality intervals.
`conf_level`	Desired confidence coverage; default `0.95`.
`alpha_lower, alpha_upper`	Optional Type I error on the lower and upper side.
`seed`	Optional integer seed for the bootstrap, for reproducibility. Used locally: the caller's random number generator state is restored on exit. Default `NULL` leaves the random number generator state alone.

Details

Why CI = "none" is the default. Confidence interval construction for η^2_G is not as settled as for partial η^2 or ω^2, because the denominator mixes sums of squares from heterogeneous sources (the focal effect, one or more measured factors, and the error term). No noncentral F transformation maps the population noncentrality parameter directly to η^2_G. Both methods below are approximations and are exposed for exploration rather than as defaults.

`method = "parametric"`**.** The function first obtains a confidence interval for the population noncentrality parameter λ of the focal effect's F-test via `ci_nc_F`. The NCP bounds are mapped through the partial-η^2 transformation $\eta^2_{p,\text{bound}} = \lambda_{\text{bound}}/(\lambda_{\text{bound}} + N)$ (matching the convention used by `ci_pvaf` and `ci_omega_squared`), and then re-expressed as η^2_G bounds via

$$\eta^2_{G,\text{bound}} = \frac{r_{\text{bound}}}{r_{\text{bound}} + r_{\text{obs}} + 1},$$

where $r_{\text{bound}} = \eta^2_{p,\text{bound}}/(1 - \eta^2_{p,\text{bound}})$ and $r_{\text{obs}} = \sum SS_{\text{measured}}/SS_{\text{error}}$. This treats the observed-factor sums of squares as fixed at their sample values, so the interval inherits whatever sampling variability those contribute. It has not been validated for coverage and should be treated as preliminary.

`method = "bootstrap"`**.** A residual bootstrap from the fitted model: the resampling unit is a residual, drawn nonparametrically from the model's own residuals rather than from a fitted distribution. For each of the `B` replications, the response is regenerated as $\hat{y}_i + \varepsilon^*_i$ where ε^* is sampled with replacement from the model's residuals; the model is refit; η^2_G is recomputed; and the percentile interval, the empirical quantiles of the `B` bootstrap values (Efron & Tibshirani, 1993), is reported. The percentile interval is the only bootstrap interval offered; there is no bias-corrected and accelerated (BCa) variant. Replicates whose refit fails are dropped, and the interval is computed from the replications that return a value. Bootstrap results vary from run to run; supply `seed` for reproducibility. Requires a single-stratum `aov`/`lm` fit. **Not yet supported for** `aovlist` **(multi-stratum / within-subjects) fits**, since the bootstrap needs to respect the subject-level correlation structure, which naive residual resampling does not. Use `method = "parametric"` for `aovlist` fits. Coverage has not been broadly validated for this estimand.

Within-subjects designs (`aovlist`**).** For multi-stratum fits the parametric CI uses each focal effect's stratum-specific F test and degrees of freedom. The denominator ratio is adjusted to reflect the full set of error strata: the implied SS_{error} for the focal effect is its own stratum's residual SS, while the r_{obs} term includes both measured-factor SS and the residual SS of all *other* strata. This generalizes the partial-η^2 CI machinery to the Bakeman (2005) denominator.

Value

A `data.frame` with one row per focal effect and the columns `effect`, `eta_squared_generalized`, `lower_limit`, `upper_limit`, and `method`. For `aovlist` fits a `stratum` column is also present. When `method = "none"`, the limit columns contain `NA`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Algina, J., Keselman, H. J., & Penfield, R. D. (2005). An alternative to Cohen's standardized mean difference effect size: A robust parameter and confidence interval in the two independent groups case. *Psychological Methods, 10*(3), 317–328. doi:10.1037/1082989X.10.3.317

Bakeman, R. (2005). Recommended effect size statistics for repeated measures designs. *Behavior Research Methods, 37*(3), 379–384. doi:10.3758/BF03192707

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, NY: Chapman & Hall/CRC.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*, 137–152. doi:10.1037/a0028086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on η^2, Chapter 7 on factorial designs, and Chapter 11 on generalized η^2 for within-subjects designs.)

Olejnik, S., & Algina, J. (2003). Generalized eta and omega squared statistics: Measures of effect size for some common research designs. *Psychological Methods, 8*(4), 434–447. doi:10.1037/1082989X.8.4.434

Smithson, M. (2001). Correct confidence intervals for various regression effect sizes and parameters: The importance of noncentral distributions in computing intervals. *Educational and Psychological Measurement, 61*, 605–632. doi:10.1177/00131640121971392

Steiger, J. H. (2004). Beyond the *F* test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

`eta_squared_generalized`, `ci_eta_squared`

Other confidence intervals for effect sizes: `ci_R2()`, `ci_c()`, `ci_c_ancova()`, `ci_c_ancova_bp()`, `ci_correlation`, `ci_cv()`, `ci_eta_squared()`, `ci_eta_squared_partial()`, `ci_mahalanobis()`, `ci_omega_squared()`, `ci_pvaf()`, `ci_rc()`, `ci_reg_coef()`, `ci_rmsea()`, `ci_sc()`, `ci_sc_ancova()`, `ci_sm()`, `ci_smd()`, `ci_smd_c()`, `ci_snr()`, `ci_src()`, `ci_srsnr()`, `contrast_adjusted()`, `plot_smd()`

Examples

```
# The pygmalion expectancy experiment: treatment is manipulated, while
# grade is a measured classification, so grade belongs in the denominator.
pyg <- pygmalion
pyg$grade <- factor(pyg$grade)
fit <- aov(iq_8 ~ treatment * grade, data = pyg)

# The default returns the point estimate and leaves the limits NA, because
# both interval methods are approximations.
ci_eta_squared_generalized(fit, observed = "grade")

# The parametric approximation maps a noncentral F interval for the focal
# effect through the observed sums of squares. Warnings mark the method as
# preliminary and report that the interaction's lower limit is clamped at
# 0; treat the limits accordingly.
ci_eta_squared_generalized(fit, observed = "grade", method = "parametric")

# The third option is a residual bootstrap, which refits the model once per
# replication. B = 1000 is the smallest count the function accepts and is
# what keeps this page quick; a reported interval deserves B = 10000 or
# more. The seed makes the interval reproducible and leaves the random
# number stream of the surrounding session as it was.
ci_eta_squared_generalized(fit, observed = "grade",
                           method = "bootstrap", B = 1000, seed = 113)

# Within-subjects ANOVA. The parametric CI uses each effect's own stratum.
set.seed(113)
n <- 20
rm_data <- data.frame(
  subject = factor(rep(seq_len(n), each = 3)),
  time    = factor(rep(c("Pre", "Mid", "Post"), n),
                   levels = c("Pre", "Mid", "Post")),
  y       = rnorm(n, sd = 1.5)[rep(seq_len(n), each = 3)] +
            0.7 * rep(1:3, n) + rnorm(n * 3, sd = 1.2)
)
fit_rm <- aov(y ~ time + Error(subject/time), data = rm_data)
ci_eta_squared_generalized(fit_rm, method = "parametric")
```

`ci_eta_squared_partial` *Confidence Interval for Partial Eta Squared (Effect Size for ANOVA)*

Description

Computes the point estimate and an exact, noncentrality-based confidence interval for the population *partial* eta squared (η_p^2). Accepts either the raw ANOVA summary (*F*, effect df, error df, total *N*) or a fitted `aov`/`lm`/`aovlist` object, in which case the function returns one row per effect (with `stratum` identification for within-subjects fits).

Usage

```
ci_eta_squared_partial(
  object = NULL,
  F_value = NULL,
  df_effect = NULL,
  df_error = NULL,
  N = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL
)
```

Arguments

`object`	Optional. A fitted model object of class `aov`, `lm`, or `aovlist`.
`F_value`	Observed F-value (ignored if `object` is supplied).
`df_effect`	Numerator degrees of freedom for the effect (ignored if `object` is supplied).
`df_error`	Error (residual) degrees of freedom (ignored if `object` is supplied).
`N`	Total sample size (ignored if `object` is supplied).
`conf_level`	Desired confidence coverage; default `0.95`.
`alpha_lower, alpha_upper`	Optional Type I error on the lower and upper side.

Details

This is the explicitly-named counterpart of `ci_eta_squared`. The two share point-estimate and CI machinery: in a one-way ANOVA partial η^2 coincides with η^2; in a factorial or within-subjects ANOVA both functions return the per-effect *partial* value computed against that effect's own error stratum. Use `ci_eta_squared_partial` when you want the function name to make the partial interpretation explicit.

Value

A `data.frame` with one row per effect. Single-stratum fits and the raw interface return columns `effect`, `eta_squared_partial`, `lower_limit`, `upper_limit`, `F_value`, `df_effect`, `df_error`, `N`. `aovlist` fits additionally include a `stratum` column. With the raw-argument interface `effect` is `"overall"`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1973). Eta-squared and partial eta-squared in fixed factor ANOVA designs. *Educational and Psychological Measurement, 33*(1), 107–112.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*, 137–152. doi:10.1037/a0028086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on η^2, Chapter 7 on factorial designs, and Chapter 11 on generalized η^2 for within-subjects designs.)

Smithson, M. (2001). Correct confidence intervals for various regression effect sizes and parameters: The importance of noncentral distributions in computing intervals. *Educational and Psychological Measurement, 61*, 605–632. doi:10.1177/00131640121971392

Steiger, J. H. (2004). Beyond the *F* test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

eta_squared_partial, ci_eta_squared

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
# Raw-argument interface.
ci_eta_squared_partial(F_value = 11.221, df_effect = 4,
                       df_error = 50, N = 55)

# Two-factor ANOVA: per-effect partial eta squared with CI for the
# manipulated expectancy treatment and the measured grade
# classification (pygmalion data, N = 310). The treatment by grade
# interaction is weak here (F = 1.19), so the additive model is used.
fit <- aov(iq_8 ~ treatment + factor(grade), data = pygmalion)
ci_eta_squared_partial(fit)

# Within-subjects ANOVA.
set.seed(113)
n <- 20
rm_data <- data.frame(
  subject = factor(rep(seq_len(n), each = 3)),
  time    = factor(rep(c("Pre", "Mid", "Post"), n),
                   levels = c("Pre", "Mid", "Post")),
  y       = rnorm(n, sd = 1.5)[rep(seq_len(n), each = 3)] +
            0.7 * rep(1:3, n) + rnorm(n * 3, sd = 1.2)
)
fit_rm <- aov(y ~ time + Error(subject/time), data = rm_data)
ci_eta_squared_partial(fit_rm)
```

`ci_games_howell` *Provides Games–Howell Simultaneous Confidence Intervals for All Pairwise Comparisons Without Assuming Homogeneity of Variance*

Description

Provides Games–Howell Simultaneous Confidence Intervals for All Pairwise Comparisons Without Assuming Homogeneity of Variance

Usage

```
ci_games_howell(x, group = NULL, conf_level = 0.95)
```

Arguments

`x`	Either (a) a fitted `lm` or `aov` object with a single factor predictor, or (b) a numeric vector of the outcome, in which case `group` must also be supplied.
`group`	When `x` is a vector, a factor (or coercible to factor) giving the group membership of each observation.
`conf_level`	Family-wise confidence level. Default `0.95`.

Details

Tukey's HSD and the Kramer modification for unequal *n* (`ci_tukey_kramer`) both pool the within-group variances into MS_W, so both assume homogeneity of variance. Neither is robust when that assumption fails. The Games–Howell procedure drops the assumption: it uses a separate error term for each pair and a Welch–Satterthwaite degrees of freedom for each pair, then takes its critical value from the studentized range.

For groups g and h, the standard error of the difference uses only those two groups' variances, and the degrees of freedom are

$$df = \frac{(s_g^2/n_g + s_h^2/n_h)^2}{s_g^4/[n_g^2(n_g-1)] + s_h^4/[n_h^2(n_h-1)]},$$

the same Welch–Satterthwaite expression that base R's `t.test` uses by default for two groups. A pair is declared different when the observed t exceeds $q/\sqrt{2}$, with q the studentized range critical value (`cv_tukey_hsd`) at the pair's degrees of freedom, so the interval for the difference of means is $(\bar{Y}_g - \bar{Y}_h) \pm q_{\alpha;a,df}\sqrt{(s_g^2/n_g + s_h^2/n_h)/2}$. Maxwell, Delaney, and Kelley (2027, Chapter 5) develop this as one of the two modifications of Tukey's HSD for heterogeneous variances (their Equations 5.13 and 5.14).

When to use it. Reach for Games–Howell when the group variances are not interchangeable and the design is between subjects. It is the heterogeneity-robust counterpart of `ci_tukey_kramer` and, like it, controls the family-wise error rate across all $a(a-1)/2$ pairs. It handles unequal *n* as a matter of course, so it does not need a separate unequal-*n* variant.

When something else is better. Dunnett (1980) found that Games–Howell becomes slightly liberal (the family-wise error rate runs somewhat above the nominal level) when the samples are small.

Maxwell, Delaney, and Kelley (2027, Chapter 5) therefore recommend Games–Howell for larger samples and Dunnett's T3, which takes its critical value from the studentized maximum modulus (cv_smm) rather than the studentized range, when the groups have fewer than roughly 50 observations each. When the variances are in fact homogeneous, use ci_tukey_kramer instead: it pools the variances, so it has more error degrees of freedom and more power. When only treatments are compared to a single control, use ci_dunnett.

With $a = 2$ groups the procedure is exactly Welch's t test: the interval and the p-value equal those from `t.test(..., var.equal = FALSE)`, because $q_{\alpha;2,df} = \sqrt{2}\, t_{1-\alpha/2,df}$.

Value

A `data.frame` with one row per pairwise comparison and columns `contrast`, `mean_difference`, `se`, `df`, `q_statistic`, `lower_limit`, `upper_limit`, and `p_adjusted`. The `df` column is the Welch–Satterthwaite degrees of freedom for that pair, which is why it varies from row to row. The table prints through the dmar_tbl display layer and works with tidy and glance (see dmar_tidiers).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Games, P. A., & Howell, J. F. (1976). Pairwise multiple comparison procedures with unequal *n*'s and/or variances: A Monte Carlo study. *Journal of Educational Statistics, 1*(2), 113–125. doi:10.2307/1164979

Dunnett, C. W. (1980). Pairwise multiple comparisons in the unequal variance case. *Journal of the American Statistical Association, 75*(372), 796–800. doi:10.2307/2287161

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 5 on the multiple-comparisons problem, where the modifications of Tukey's HSD for unequal *n* and unequal variances are developed.)

See Also

ci_tukey_kramer for the homogeneity-assuming counterpart, ci_dunnett for many-to-one comparisons, ci_scheffe for arbitrary contrasts, cv_tukey_hsd for the critical value it uses, and dmar_tidiers for the tidy methods.

Examples

```
# The raw drinks_per_week outcome of the drinks_trial data: the
# Standard arm's variance is more than three times that of either CRA
# arm, so the pooled error term Tukey's HSD relies on is questionable.
ci_games_howell(drinks_trial$drinks_per_week, drinks_trial$treatment)

# A fitted one-way model may be passed instead of the two vectors.
ci_games_howell(aov(drinks_per_week ~ treatment, data = drinks_trial))

# With two groups the procedure is Welch's t test, so the limits agree;
```

```
# the trial's two enrollment cohorts give a two-group comparison.
ci_games_howell(drinks_trial$drinks_per_week, drinks_trial$cohort)$lower_limit
-t.test(drinks_per_week ~ cohort, data = drinks_trial)$conf.int[2]
```

`ci_mahalanobis` *Confidence Interval for the Squared Mahalanobis Distance*

Description

Computes the squared Mahalanobis distance D^2 together with an exact confidence interval for the population squared distance Δ^2, obtained by inverting Hotelling's T^2 statistic through its (noncentral) F-distribution as in Reiser (2001). Both the one-sample setting (mean vector against a hypothesized population mean) and the two-sample setting (between-groups distance from discriminant analysis or multivariate group comparison) are supported, and either raw data or a pre-computed D^2 with sample sizes can be supplied.

Usage

```
ci_mahalanobis(
  D2 = NULL,
  group_1 = NULL,
  group_2 = NULL,
  mu_0 = NULL,
  n_1 = NULL,
  n_2 = NULL,
  p = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  ...
)
```

Arguments

`D2`	Optional pre-computed squared Mahalanobis distance. Ignored if `group_1` is supplied.
`group_1`	Optional numeric matrix or data frame for the first sample ($n_1 \times p$). Rows are observations and columns are variables.
`group_2`	Optional numeric matrix or data frame for the second sample ($n_2 \times p$). When `NULL`, the function operates in one-sample mode against `mu_0`.
`mu_0`	Optional hypothesized population mean for the one-sample case (length-p numeric vector). Defaults to a vector of zeros.
`n_1`	Sample size for group 1 (required when supplying `D2` directly).
`n_2`	Sample size for group 2 (required for two-sample mode when supplying `D2` directly; leave `NULL` for one-sample mode).

`p` Dimensionality (number of variables) when supplying `D2` directly.

`conf_level` Confidence coverage for a symmetric interval (default `0.95`). Set it to `NULL` to specify the tails directly through `alpha_lower` and `alpha_upper`.

`alpha_lower`, `alpha_upper` Optional Type I error rates for the lower and upper tail. To use them, set `conf_level = NULL` and supply both (an asymmetric or one-sided interval with coverage `1 - alpha_lower - alpha_upper`); supplying either alongside a non-`NULL` `conf_level` is an error, as in `ci_nc_F`, to which they are passed.

`...` Additional arguments passed to `ci_nc_F` (for example `tol`).

Details

Definition. For a p-vector $\mathbf{x}$ drawn from a multivariate normal with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, Mahalanobis's (1936) squared distance from a reference vector $\boldsymbol{\mu}_0$ is

$$\Delta^2 = (\boldsymbol{\mu} - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}_0).$$

In the two-sample case the population distance between groups is $\Delta^2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$, assuming a common covariance. The corresponding sample estimates plug the sample means and the sample (or pooled) covariance into the same quadratic form.

Link to Hotelling's T^2. Hotelling's (1931) T^2 statistic is $T^2 = nD^2$ (one sample) or $T^2 = \{n_1 n_2/(n_1 + n_2)\} D^2$ (two samples). Under multivariate normality

$$\frac{n_1 + n_2 - p - 1}{(n_1 + n_2 - 2)\, p}\, T^2 \sim F'\left(p,\, n_1 + n_2 - p - 1,\, \lambda = \frac{n_1 n_2}{n_1 + n_2}\, \Delta^2\right)$$

in the two-sample case, and analogously $\{(n - p)/[(n - 1)p]\}\, T^2 \sim F'(p, n - p, n\Delta^2)$ in the one-sample case (see Anderson, 2003, Section 5.2).

Confidence interval. The CI on Δ^2 is obtained by inverting these distributional results (Reiser, 2001): a CI on the noncentrality parameter λ is constructed via `ci_nc_F` and then mapped back to Δ^2 by $\Delta^2 = \lambda\,(n_1 + n_2)/(n_1 n_2)$ (two sample) or $\Delta^2 = \lambda/n$ (one sample). When the observed F is below the lower-tail critical value of the central F-distribution at the requested confidence level, the lower CI on λ (and hence on Δ^2) is clamped to zero, in keeping with the `ci_nc_F` convention.

Bias. The plug-in estimator D^2 is upward biased for Δ^2; the CI from this function is exact for Δ^2 under multivariate normality and reflects the bias structure correctly, but the point estimate reported is the standard plug-in D^2.

Value

A one-row `data.frame` with columns `sample_type` (`"one-sample"` or `"two-sample"`), `D2` (point estimate of the squared distance), `lower_limit` and `upper_limit` (the confidence limits on the population squared distance Δ^2), `F_value`, `df_1`, `df_2`, `n_1`, `n_2` (`NA` in one-sample mode), and `p`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis* (4th ed.). Wiley.

Hotelling, H. (1931). The generalization of Student's ratio. *The Annals of Mathematical Statistics, 2*(3), 360–378.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Sciences of India, 2*(1), 49–55.

Reiser, B. (2001). Confidence intervals for the Mahalanobis distance. *Communications in Statistics–Simulation and Computation, 30*(1), 37–45. doi:10.1081/SAC100001856

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

See Also

ci_nc_F, ci_smd, ci_R2

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
# Two-sample distance between the two schools of the Holzinger and
#     Swineford (1939) study on a four-test cognitive battery.
battery <- c("t1_visual_perception", "t2_cubes", "t4_lozenges",
             "t6_paragraph_comprehension")
g1 <- as.matrix(holzinger_swineford[
  holzinger_swineford$school == "Grant-White", battery])
g2 <- as.matrix(holzinger_swineford[
  holzinger_swineford$school == "Pasteur", battery])
ci_mahalanobis(group_1 = g1, group_2 = g2)

# One-sample distance: how far is the Grant-White centroid from a
#     reference vector of (29, 24, 18, 9)?
ci_mahalanobis(group_1 = g1, mu_0 = c(29, 24, 18, 9))

# Pre-computed D^2 (no raw data needed): the two-school distance,
#     reproduced from reported summaries.
ci_mahalanobis(D2 = 0.608, n_1 = 145, n_2 = 156, p = 4)
```

`ci_nc_chisq` *Confidence Interval for the Noncentrality Parameter of a Noncentral Chi Square Distribution*

Description

Finds the noncentrality parameters of a noncentral chi square distribution that bracket an observed chi square value with the requested tail probabilities, giving a confidence interval on the population noncentrality parameter. Together with `ci_nc_t` and `ci_nc_F`, this is one of the low-level noncentral distribution workhorses on which the `ci_*` confidence interval functions are built; most analyses reach it through those functions rather than calling it directly. The function was `conf_limits_nc_chisq()` in earlier builds of DMAR and is `conf.limits.nc.chisq()` in MBESS; it is named into the `ci_*` family because a confidence interval is what it computes.

Usage

```
ci_nc_chisq(
  chi_square = NULL,
  conf_level = 0.95,
  df = NULL,
  alpha_lower = NULL,
  alpha_upper = NULL,
  tol = 1e-09,
  verbose = TRUE,
  ...
)
```

Arguments

`chi_square`	The observed chi square value
`conf_level`	The desired degree of confidence for a symmetric interval
`df`	The degrees of freedom
`alpha_lower`	The proportion of values beyond the lower limit (cannot be used with `conf_level`)
`alpha_upper`	The proportion of values beyond the upper limit (cannot be used with `conf_level`)
`tol`	The convergence tolerance passed to `uniroot`
`verbose`	If `TRUE` (the default), the returned data frame additionally reports the achieved tail probabilities at each limit; if `FALSE`, only `term` and `value` are returned
`...`	Additional arguments forwarded to `uniroot`

Details

Each confidence limit is the noncentrality parameter $\lambda \geq 0$ of a noncentral chi square distribution with `df` degrees of freedom whose appropriate tail at the observed `chi_square` contains the requested probability:

- the lower limit satisfies $P(X \geq \texttt{chi\_square}) = \texttt{alpha\_lower}$;
- the upper limit satisfies $P(X \leq \texttt{chi\_square}) = \texttt{alpha\_upper}$.

The two conditions run in opposite directions in λ: the lower-tail probability $P(X \leq \texttt{chi\_square})$ is continuous and strictly decreasing in the noncentrality parameter, so the upper-tail probability $P(X \geq \texttt{chi\_square})$ is continuous and strictly increasing in it. The lower limit is the λ at which the upper tail has grown to `alpha_lower`, and the upper limit is the λ at which the lower tail has shrunk to `alpha_upper`. Each is therefore the unique non-negative root of a one-dimensional equation, and both are located with `uniroot` on the decreasing lower-tail scale; `extendInt` is used to widen the search bracket if needed.

Because the noncentrality parameter is bounded below by zero, the lower limit is set to zero whenever the observed `chi_square` is smaller than the `alpha_lower` critical value of the central chi square distribution (i.e., the data is consistent with $\lambda = 0$ at the requested confidence level). A warning is issued in that case, and the achieved probabilities reported in the output reflect the actual values at $\lambda = 0$.

Symmetrically, when the observed `chi_square` is so small that even at $\lambda = 0$ the lower-tail probability is already at or below `alpha_upper`, no $\lambda \geq 0$ places as much as `alpha_upper` mass at or below `chi_square`; the upper limit is undefined and is returned as `NA`, with a warning.

Value

A `data.frame` with one row per confidence limit and the columns:

`term`	Either `"lower_limit"` or `"upper_limit"`.
`value`	The noncentrality parameter at that limit. `0` when `alpha_lower = 0` or the lower limit is unattainable; `Inf` when `alpha_upper = 0`; `NA` when the observed `chi_square` is so small that even at $\lambda = 0$ the lower-tail probability is already at or below `alpha_upper`, so the upper limit is undefined (a warning is issued).
`prob_less`	(`verbose = TRUE`) The probability $P(X \leq \texttt{chi\_square})$ that an observation from the noncentral chi square distribution centered at the row's limit falls at or below the observed `chi_square`.
`prob_greater`	(`verbose = TRUE`) The complementary probability $P(X \geq \texttt{chi\_square})$. By construction this equals `alpha_lower` on the `lower_limit` row and $1 - \texttt{alpha\_upper}$ on the `upper_limit` row.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*(4), 532–574. doi:10.1177/0013164401614002

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

See Also

ci_nc_t, ci_nc_F, stats::pchisq(), stats::qchisq(), uniroot

Other noncentral distribution confidence intervals: ci_nc_F(), ci_nc_t()

Examples

```
# A typical call to the function.
ci_nc_chisq(chi_square = 30, conf_level = .95, df = 15)

# A one-sided (upper) confidence interval.
ci_nc_chisq(chi_square = 30, alpha_lower = 0, alpha_upper = .05,
                     conf_level = NULL, df = 15)
```

---

ci_nc_F *Confidence Interval for the Noncentrality Parameter of a Noncentral* F *Distribution*

Description

Finds the noncentrality parameters of a noncentral F-distribution that bracket an observed F-value with the requested tail probabilities, giving a confidence interval on the population noncentrality parameter. Together with ci_nc_t and ci_nc_chisq, this is one of the low-level noncentral distribution workhorses on which the ci_* confidence interval functions (e.g., ci_pvaf, ci_snr, ci_R2) are built; most analyses reach it through those functions rather than calling it directly. The function was conf_limits_ncf() in earlier builds of DMAR and is conf.limits.ncf() in MBESS; it is named into the ci_* family because a confidence interval is what it computes.

Usage

```
ci_nc_F(
  F_value = NULL,
  conf_level = 0.95,
  df_1 = NULL,
  df_2 = NULL,
  alpha_lower = NULL,
  alpha_upper = NULL,
  tol = 1e-09,
  verbose = TRUE,
  ...
)
```

Arguments

F_value	The observed F-value
conf_level	The desired degree of confidence for a symmetric interval
df_1	The numerator degrees of freedom

df_2	The denominator degrees of freedom
alpha_lower	The proportion of values beyond the lower limit (cannot be used with conf_level)
alpha_upper	The proportion of values beyond the upper limit (cannot be used with conf_level)
tol	The convergence tolerance passed to uniroot
verbose	If TRUE (the default), the returned data frame additionally reports the achieved tail probabilities at each limit; if FALSE, only term and value are returned
...	Additional arguments forwarded to uniroot

Details

Each confidence limit is the noncentrality parameter $\lambda \geq 0$ of a noncentral F-distribution with df_1 and df_2 degrees of freedom whose appropriate tail at the observed F_value contains the requested probability:

- the lower limit satisfies $P(F \geq \texttt{F\_value}) = \texttt{alpha\_lower}$;
- the upper limit satisfies $P(F \leq \texttt{F\_value}) = \texttt{alpha\_upper}$.

The two conditions run in opposite directions in λ: the lower-tail probability $P(F \leq \texttt{F\_value})$ is continuous and strictly decreasing in the noncentrality parameter, so the upper-tail probability $P(F \geq \texttt{F\_value})$ is continuous and strictly increasing in it. The lower limit is the λ at which the upper tail has grown to alpha_lower, and the upper limit is the λ at which the lower tail has shrunk to alpha_upper. Each is therefore the unique non-negative root of a one-dimensional equation, and both are located with uniroot on the decreasing lower-tail scale; extendInt is used to widen the search bracket if needed.

Because the noncentrality parameter is bounded below by zero, the lower limit is set to zero whenever the observed F_value is smaller than the alpha_lower critical value of the central F-distribution (i.e., the data is consistent with $\lambda = 0$ at the requested confidence level). A warning is issued in that case, and the achieved probabilities reported in the output reflect the actual values at $\lambda = 0$ rather than the requested alpha_lower. The warning carries the condition class dmar_nc_F_clamp, so a caller that inverts the noncentral F repeatedly can muffle or deduplicate it by class.

Value

A data.frame with one row per confidence limit and the columns:

term	Either "lower_limit" or "upper_limit".
value	The noncentrality parameter at that limit. 0 when alpha_lower = 0 or the lower limit is unattainable; Inf when alpha_upper = 0; NA on the upper_limit row when the observed F_value is so small that even at $\lambda = 0$ the lower-tail probability is already at or below alpha_upper, leaving the upper noncentrality limit undefined (a warning is issued).
prob_less	(verbose = TRUE) The probability $P(F \leq \texttt{F\_value})$ that an F-statistic from the noncentral F-distribution centered at the row's limit falls at or below the observed F_value.
prob_greater	(verbose = TRUE) The complementary probability $P(F \geq \texttt{F\_value})$. By construction this equals alpha_lower on the lower_limit row and $1-$alpha_upper on the upper_limit row.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*(4), 532–574. doi:10.1177/0013164401614002

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

See Also

ss_aipe_R2, ci_R2, ci_nc_t, ci_nc_chisq, stats::pf(), stats::qf(), uniroot

Other noncentral distribution confidence intervals: ci_nc_chisq(), ci_nc_t()

Examples

```
ci_nc_F(F_value = 5, conf_level = .95, df_1 = 5, df_2 = 100)

# A one-sided (upper) confidence interval.
ci_nc_F(F_value = 5, conf_level = NULL, df_1 = 5, df_2 = 100,
                alpha_lower = 0, alpha_upper = .05)
```

---

ci_nc_t — *Confidence Interval for the Noncentrality Parameter of a Noncentral* t *Distribution*

Description

Finds the noncentrality parameters of a noncentral *t*-distribution that bracket an observed *t*-value with the requested tail probabilities, giving a confidence interval on the population noncentrality parameter. Together with ci_nc_F and ci_nc_chisq, this is one of the low-level noncentral distribution workhorses on which the ci_* confidence interval functions (e.g., ci_smd, ci_smd_c, ci_cv) are built; most analyses reach it through those functions rather than calling it directly. The function was conf_limits_nct() in earlier builds of DMAR and is conf.limits.nct() in MBESS; it is named into the ci_* family because a confidence interval is what it computes.

Usage

```
ci_nc_t(
  ncp,
  df,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
```

```
  t_value,
  tol = 1e-09,
  verbose = TRUE,
  ...
)
```

Arguments

ncp	The noncentrality parameter (e.g., observed *t*-value) of interest
df	The degrees of freedom
conf_level	The level of confidence for a symmetric confidence interval
alpha_lower	The proportion of values beyond the lower limit of the confidence interval (cannot be used with conf_level)
alpha_upper	The proportion of values beyond the upper limit of the confidence interval (cannot be used with conf_level)
t_value	Alias for ncp
tol	The convergence tolerance passed to uniroot when locating each limit
verbose	If TRUE (the default), the returned data frame additionally reports the achieved tail probabilities at each limit; if FALSE, only term and value are returned
...	Additional arguments forwarded to uniroot

Details

Each confidence limit is the noncentrality parameter of a noncentral *t*-distribution with df degrees of freedom whose appropriate tail at the observed ncp contains the requested probability:

- the lower limit satisfies $P(T \geq \text{ncp}) = \texttt{alpha\_lower}$;
- the upper limit satisfies $P(T \leq \text{ncp}) = \texttt{alpha\_upper}$.

Each tail probability is continuous and strictly monotone in the noncentrality parameter, so each limit is the unique root of a one-dimensional equation. The roots are located with uniroot starting from a bracket centered on ncp with half-width scaled by the asymptotic standard error of the noncentrality estimator; extendInt is used to widen the bracket if needed.

This function is especially useful for forming confidence intervals around standardized mean differences (Cohen's *d*, Glass's *g*, Hedges' *g*), standardized regression coefficients, and coefficients of variation.

Value

A data.frame with one row per confidence limit and the columns:

term	Either "lower_limit" or "upper_limit".
value	The noncentrality parameter at that limit. -Inf when alpha_lower = 0; Inf when alpha_upper = 0.
prob_less	(verbose = TRUE) The probability $P(T \leq \text{ncp})$ that a *t*-statistic from the noncentral *t*-distribution centered at the row's limit falls at or below the observed ncp. By construction this equals alpha_upper on the upper_limit row and $1 -$ alpha_lower on the lower_limit row.

prob_greater (verbose = TRUE) The complementary probability $P(T \geq \text{ncp})$. By construction this equals alpha_lower on the lower_limit row and $1 -$ alpha_upper on the upper_limit row.

Warning

As of R 4.0.0, the largest ncp that R can accurately handle is 37.62.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*(4), 532–574. doi:10.1177/0013164401614002

Kelley, K. (2005). The effects of nonnormal distributions on confidence intervals around the standardized mean difference: Bootstrap and parametric confidence intervals, *Educational and Psychological Measurement, 65*, 51–69. doi:10.1177/0013164404264850

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

stats::pt(), stats::qt(), uniroot, ci_smd, ci_smd_c, ci_nc_F, ci_nc_chisq

Other noncentral distribution confidence intervals: ci_nc_F(), ci_nc_chisq()

Examples

```
# Suppose observed t-value based on 'df'=126 is 2.83. Finding the lower
# and upper critical values for the population noncentrality parameter
# with a symmetric confidence interval with 95\% confidence is given as:
ci_nc_t(ncp = 2.83, df = 126, conf_level = .95)

# Modifying the above example so that a nonsymmetric 95% confidence interval
# can be formed:
ci_nc_t(ncp = 2.83, df = 126, alpha_lower = .01, alpha_upper = .04, conf_level = NULL)

# Modifying the above example so that a single-sided 95% confidence interval
# can be formed:
ci_nc_t(ncp = 2.83, df = 126, alpha_lower = 0, alpha_upper = .05, conf_level = NULL)
```

`ci_omega_squared` *Confidence Interval for Omega Squared (Effect Size for ANOVA)*

Description

Computes the point estimate and an exact, noncentrality-based confidence interval for the population omega squared (ω^2), the proportion of variance in the dependent variable accounted for by a fixed effect. Accepts either the raw ANOVA summary (F, effect df, error df, total N) or a fitted `aov` / `lm` object, in which case the function returns a row per effect (partial ω^2 in factorial designs).

Usage

```
ci_omega_squared(
  object = NULL,
  F_value = NULL,
  df_effect = NULL,
  df_error = NULL,
  N = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL
)
```

Arguments

`object`	Optional. A fitted `aov` or `lm` object. When supplied, the function loops over the non-`Residuals` rows of `anova(object)` and returns one row per effect.
`F_value`	Observed F-value from the fixed-effects ANOVA (ignored if `object` is supplied).
`df_effect`	Numerator degrees of freedom for the effect (ignored if `object` is supplied).
`df_error`	Error (residual) degrees of freedom (ignored if `object` is supplied).
`N`	Total sample size (ignored if `object` is supplied; `nobs(object)` is used instead).
`conf_level`	Desired confidence coverage; default `0.95`. Used only when `alpha_lower` and `alpha_upper` are both `NULL`.
`alpha_lower, alpha_upper`	Optional Type I error on the lower and upper side. If both are `NULL`, a symmetric interval at `conf_level` is used. If both are supplied, `conf_level` is recomputed as `1 - alpha_lower - alpha_upper`.

Details

Point estimate. The function reports the usual sample omega squared, which for a one-way design can be written as

$$\hat{\omega}^2 = \frac{SS_{\text{effect}} - df_{\text{effect}} \cdot MS_{\text{error}}}{SS_{\text{total}} + MS_{\text{error}}} = \frac{df_{\text{effect}}(F-1)}{df_{\text{effect}}(F-1) + N}$$

(Hays, 1994; Keppel, 1991). For factorial designs the same formula applied per effect yields *partial* omega squared (Olejnik & Algina, 2003); values below zero are truncated to zero.

Confidence interval. The CI is constructed by Steiger's (2004, Proposition 1) confidence interval transformation principle: a CI for the noncentrality parameter λ of the F distribution is obtained (via `ci_nc_F`) and then mapped through

$$\omega^2_{\text{bound}} = \frac{\lambda_{\text{bound}}}{\lambda_{\text{bound}} + N}.$$

When the lower CI on λ is not identified (i.e., the observed F is below the one-sided critical value), the lower limit on ω^2 is set to 0, matching the convention used in `ci_pvaf`. In a one-way design, the interval produced here is identical to the CI for η^2 from `ci_pvaf`; the two estimands coincide in the population and differ only in their *sample* estimators (an implication of the confidence interval transformation principle of Steiger, 2004).

Sums of squares in factorial designs. When a fitted model is supplied, the function reads the F-values from `anova()`, which in base R uses Type I (sequential) sums of squares. For balanced designs, Types I, II, and III give identical F-values; for unbalanced designs they differ. If Type II or III F-values are required, compute them with e.g.\ `car::Anova(object, type = 3)` and pass the relevant F / df into the raw-argument interface.

Value

A `data.frame` with one row per effect and the columns `effect`, `omega_squared` (point estimate), `lower_limit`, `upper_limit`, `F_value`, `df_effect`, `df_error`, and `N`. When the raw-argument interface is used, `effect` is `"overall"`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Fleishman, A. I. (1980). Confidence intervals for correlation ratios. *Educational and Psychological Measurement, 40*(3), 659–670.

Hays, W. L. (1994). *Statistics* (5th ed.). Fort Worth, TX: Harcourt Brace College Publishers.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*, 137–152. doi:10.1037/a0028086

Keppel, G. (1991). *Design and analysis: A researcher's handbook* (3rd ed.). Prentice Hall.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on η^2, Chapter 7 on factorial designs, and Chapter 11 on generalized η^2 for within-subjects designs.)

Olejnik, S., & Algina, J. (2003). Generalized eta and omega squared statistics: Measures of effect size for some common research designs. *Psychological Methods, 8*(4), 434–447. doi:10.1037/1082989X.8.4.434

Steiger, J. H. (2004). Beyond the *F* test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

ci_pvaf, ci_nc_F

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
# 1. Raw-argument interface. Bargman's (1970) example, also used in
#        Venables (1975), Fleishman (1980), and Steiger (2004): a 5-group
#        one-way ANOVA with 11 subjects per group, observed F = 11.221.
ci_omega_squared(F_value = 11.221, df_effect = 4, df_error = 50, N = 55)

# Same example with a 90% confidence interval.
ci_omega_squared(
  F_value = 11.221, df_effect = 4, df_error = 50, N = 55,
  conf_level = 0.90
)

# 2. One way ANOVA from a fitted model: mean IQ gain differs across
#        the six grades of the pygmalion data (N = 310).
fit_one <- aov(iq_gain ~ factor(grade), data = pygmalion)
ci_omega_squared(fit_one)

# 3. Two-factor ANOVA: partial omega squared per effect for the
#        manipulated expectancy treatment and the measured grade
#        classification (pygmalion data, N = 310). The treatment by
#        grade interaction is weak here (F = 1.19), so the additive
#        model is used.
fit_additive <- aov(iq_8 ~ treatment + factor(grade), data = pygmalion)
ci_omega_squared(fit_additive)
```

---

ci_proportion *Confidence Interval for a Single Proportion*

Description

The Wilson (1927) score interval for a binomial proportion, the package's default for proportion inference: unlike the textbook Wald interval it cannot escape [0, 1], behaves sensibly at 0 and 1 counts, and holds close to nominal coverage at small *n* (Brown, Cai, & DasGupta, 2001, recommend it for general use). The Wald interval is available for instruction and comparison.

Usage

```
ci_proportion(successes, n, conf_level = 0.95, method = c("wilson", "wald"))
```

Arguments

successes	Number of successes, a single non-negative integer.
n	Number of trials, a single positive integer at least successes.
conf_level	Confidence level. Defaults to 0.95.
method	"wilson" (default) or "wald".

Value

A data.frame (class dmar_tbl) with rows lower_limit, proportion, upper_limit, successes, and n, so the point estimate sits between its confidence limits.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Brown, L. D., Cai, T. T., & DasGupta, A. (2001). Interval estimation for a binomial proportion. *Statistical Science, 16*(2), 101–133. doi:10.1214/ss/1009213286

Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association, 22*(158), 209–212.

See Also

responder_analysis, which uses this interval for each group's responder proportion.

Examples

```
ci_proportion(successes = 17, n = 50)

# The Wilson interval stays inside [0, 1] even at the boundary.
ci_proportion(successes = 0, n = 20)
```

ci_pvaf *Confidence Interval for the Proportion of Variance Accounted for (in the Dependent Variable by Knowing the Levels of the Factor)*

Description

Computes the exact confidence limits for the proportion of variance in the dependent variable accounted for by knowing the levels of the factor (group status in a single factor design) in a fixed effects analysis of variance, so an omnibus *F*-test is accompanied by an effect size with a statement of its precision.

Usage

```
ci_pvaf(
  F_value = NULL,
  df_1 = NULL,
  df_2 = NULL,
  N = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  ...
)
```

Arguments

F_value	Observed *F*-value from fixed effects analysis of variance
df_1	Numerator degrees of freedom
df_2	Denominator degrees of freedom
N	Sample size
conf_level	Confidence interval coverage (i.e., 1-Type I error rate); default is .95
alpha_lower	Type I error for the lower confidence limit
alpha_upper	Type I error for the upper confidence limit
...	Allows one to potentially include parameter values for inner functions

Details

The confidence level must be specified in one of following two ways: using confidence interval coverage (conf_level), or lower and upper confidence limits (alpha_lower and alpha_upper).

This function uses the confidence interval transformation principle (Steiger, 2004) to transform the confidence limits for the noncentrality parameter to the confidence limits for the population proportion of variance accounted for by knowing the group status. The confidence interval for the noncentral *F* parameter can be obtained from the function ci_nc_F, which is used within this function.

Value

A 4-row `data.frame` with columns `term`, `value`, `prob_less`, and `prob_greater`. The `term` values are `"lower_limit"` (the lower confidence limit on the proportion of variance accounted for, on the [0, 1] scale), `"pvaf"` (the sample proportion of variance accounted for, `df_1 * F_value / (df_1 * F_value + df_2)`, the same value that eta squared reports, so the point estimate sits between its confidence limits), `"upper_limit"` (the upper confidence limit), and `"actual_coverage"` (the achieved coverage probability, which equals `conf_level` when both tail targets are met). The `prob_less` and `prob_greater` columns report the achieved tail-error probabilities at the two limits; `NA` on the `"pvaf"` and `"actual_coverage"` rows.

Note

This function can be used for single or factorial ANOVA designs.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Fleishman, A. I. (1980). Confidence intervals for correlation ratios. *Educational and Psychological Measurement, 40*(3), 659–670.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*, 524–555. doi:10.1080/00273170802490632

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on R^2 as a model comparison effect size.)

Steiger, J. H. (2004). Beyond the *F* Test: Effect size confidence intervals and tests of close fit in the Analysis of Variance and Contrast Analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

`ci_nc_F`

Other confidence intervals for effect sizes: `ci_R2()`, `ci_c()`, `ci_c_ancova()`, `ci_c_ancova_bp()`, `ci_correlation`, `ci_cv()`, `ci_eta_squared()`, `ci_eta_squared_generalized()`, `ci_eta_squared_partial()`, `ci_mahalanobis()`, `ci_omega_squared()`, `ci_rc()`, `ci_reg_coef()`, `ci_rmsea()`, `ci_sc()`, `ci_sc_ancova()`, `ci_sm()`, `ci_smd()`, `ci_smd_c()`, `ci_snr()`, `ci_src()`, `ci_srsnr()`, `contrast_adjusted()`, `plot_smd()`

Examples

```
## Bargman (1970) gave an example in which a 5-group ANOVA with 11 subjects in each
## group is conducted and the observed F value is 11.221. This example was used
## in Venables (1975), Fleishman (1980), and Steiger (2004). If one wants to calculate the
```

```
## exact confidence interval for the proportion of variance accounted for in that example,
## this function can be used.
ci_pvaf(F_value = 11.221, df_1 = 4, df_2 = 50, N = 55)

ci_pvaf(F_value = 11.221, df_1 = 4, df_2 = 50, N = 55, conf_level = .90)

ci_pvaf(F_value = 11.221, df_1 = 4, df_2 = 50, N = 55, alpha_lower = 0, alpha_upper = .05)
```

---

ci_R2 *Confidence Interval for the Population Squared Multiple Correlation Coefficient*

Description

Constructs a confidence interval for the population squared multiple correlation coefficient ρ^2 by inverting the sampling distribution of the sample R^2. The confidence interval is for the population value ρ^2; the required input is the corresponding sample value, the observed sample squared multiple correlation coefficient R^2 (or, equivalently, the observed F-statistic and degrees of freedom). The right choice of sampling distribution, and so the right interval, depends on whether the predictors are treated as random draws from a joint multivariate normal distribution (the default and the typical case in the behavioral, educational, and social sciences) or as fixed by design (planned dosing levels, factorial covariates, etc.). The function selects the sampling distribution via `random_predictors` and inverts the corresponding noncentral distribution; the construction is the regression analogue of a noncentral distribution based CI on a standardized effect size (Steiger & Fouladi, 1992; Kelley, 2007).

Usage

```
ci_R2(
  R2 = NULL,
  df_1 = NULL,
  df_2 = NULL,
  conf_level = 0.95,
  random_predictors = TRUE,
  F_value = NULL,
  N = NULL,
  p = NULL,
  alpha_lower = NULL,
  alpha_upper = NULL,
  tol = 1e-09
)
```

Arguments

`R2`	Observed value of the sample squared multiple correlation coefficient
`df_1`	Numerator degrees of freedom

df_2	Denominator degrees of freedom
conf_level	Confidence interval coverage; 1-Type I error rate
random_predictors	Whether or not the predictor variables are random or fixed (random is default)
F_value	Obtained F-value
N	Sample size
p	Number of predictors
alpha_lower	Type I error for the lower confidence limit
alpha_upper	Type I error for the upper confidence limit
tol	The convergence tolerance passed to uniroot when random_predictors = FALSE and the confidence limits are found by inverting the noncentral F distribution (see ci_nc_F); ignored when random_predictors = TRUE, where the Lee (1971) bisection uses its own fixed tolerance

Details

Fixed vs.\ random predictors. The two regression models give *different* sampling distributions for the sample R^2, and so different confidence intervals. Under fixed predictors the design matrix is treated as constant in hypothetical replications of the study, and the omnibus F-statistic $F = (R^2/p)/((1-R^2)/(N-p-1))$ follows a noncentral F with p and $N-p-1$ degrees of freedom and noncentrality $\lambda = N\rho^2/(1-\rho^2)$ (Cohen, 1988); the CI is obtained by inverting that distribution at the supplied confidence level (see ci_nc_F). Under random predictors the design matrix is itself a draw from a joint multivariate normal distribution and the unconditional sampling distribution of the sample R^2 is given by Lee (1971); ci_R2 uses the Lee (1971) bisection (the same construction Algina and Olejnik 2000 implemented in SAS) to invert that distribution. Gatsonis and Sampson (1989) document the comparison and show that treating random predictors as fixed tends to overstate precision (and so under-state the CI width); the discrepancy is modest at moderate to large N but non-trivial at small N with moderate-to-large effects. In the behavioral, educational, and social sciences predictor variables are almost always random, so the default is random_predictors = TRUE; pass random_predictors = FALSE for designs in which the predictor variables are fixed by design.

Value

A 3-row data.frame with columns term, value, prob_less, and prob_greater. The rows are ordered "lower_limit" (lower confidence limit on the population ρ^2), "R2" (the sample squared multiple correlation coefficient supplied by the user, the point estimate), and "upper_limit" (upper confidence limit on the population ρ^2), so the point estimate sits between its confidence limits. The prob_less and prob_greater columns report the achieved lower-tail and upper-tail error probabilities at each limit (they are NA for the "R2" estimate row). For random-predictor mode (random_predictors = TRUE) the limits are computed via the Lee (1971) bisection over the multiple-correlation sampling distribution; for fixed-predictor mode they are computed by inversion of the noncentral F distribution.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Algina, J. & Olejnik, S. (2000). Determining sample size for accurate estimation of the squared multiple correlation coefficient. *Multivariate Behavioral Research, 35*, 119–137. doi:10.1207/s15327906mbr3501_5

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Gatsonis, C., & Sampson, A. R. (1989). Multiple correlation: Exact power and sample size calculations. *Psychological Bulletin, 106*(3), 516–524.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*, 524–555. doi:10.1080/00273170802490632

Lee, Y. S. (1971). Some results on the sampling distribution of the multiple correlation coefficient. *Journal of the Royal Statistical Society, Series B, 33*(1), 117–130.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on R^2 as a model comparison effect size.)

Smithson, M. (2003). *Confidence intervals*. Thousand Oaks, CA: Sage Publications.

Steiger, J. H., & Fouladi, R. T. (1992). R2: A computer program for interval estimation, power calculations, sample size estimation, and hypothesis testing in multiple regression. *Behavior Research Methods, Instruments, & Computers, 24*(4), 581–582. doi:10.3758/BF03203611

See Also

ss_aipe_R2, ci_nc_F

Other confidence intervals for effect sizes: ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
# For random predictor variables.
ci_R2(R2 = .25, N = 100, p = 5, conf_level = .95, random_predictors = TRUE)

ci_R2(F_value = 6.266667, N = 100, p = 5, conf_level = .95, random_predictors = TRUE)

# For fixed predictor variables.
ci_R2(R2 = .25, N = 100, p = 5, conf_level = .95, random_predictors = FALSE)

ci_R2(F_value = 6.266667, N = 100, p = 5, conf_level = .95, random_predictors = FALSE)

# One sided confidence intervals when predictors are random.
ci_R2(R2 = .25, N = 100, p = 5, alpha_lower = .05, alpha_upper = 0,
      conf_level = NULL, random_predictors = TRUE)
```

```
ci_R2(R2 = .25, N = 100, p = 5, alpha_lower = 0, alpha_upper = .05,
      conf_level = NULL, random_predictors = TRUE)

# One sided confidence intervals when predictors are fixed.
ci_R2(R2 = .25, N = 100, p = 5, alpha_lower = .05, alpha_upper = 0,
      conf_level = NULL, random_predictors = FALSE)

ci_R2(R2 = .25, N = 100, p = 5, alpha_lower = 0, alpha_upper = .05,
      conf_level = NULL, random_predictors = FALSE)
```

`ci_rc` *Confidence Interval for an Unstandardized Regression Coefficient*

Description

Computes a confidence interval for a population regression coefficient in its raw (unstandardized) metric, by the standard *t*-based approach or the noncentral *t* approach. A thin convenience wrapper around `ci_reg_coef`, which is the general engine; for the standardized coefficient use `ci_src`.

Usage

```
ci_rc(
  b_j,
  SE_b_j = NULL,
  s_Y = NULL,
  s_X = NULL,
  N,
  p,
  R2_Y_X = NULL,
  R2_j_X_without_j = NULL,
  conf_level = 0.95,
  R2_Y_X_without_j = NULL,
  t_value = NULL,
  alpha_lower = NULL,
  alpha_upper = NULL,
  noncentral = FALSE,
  ...
)
```

Arguments

`b_j`	Value of the regression coefficient for the *j*th predictor variable
`SE_b_j`	Standard error for the *j*th predictor variable
`s_Y`	Standard deviation of *Y*, the dependent variable
`s_X`	Standard deviation of *X*, the predictor variable of interest

N	Sample size
p	The number of predictors
R2_Y_X	The squared multiple correlation coefficient predicting *Y* from the *p* predictor variables
R2_j_X_without_j	The squared multiple correlation coefficient predicting the *j*th predictor variable (i.e., the predictor of interest) from the remaining *p*-1 predictor variables
conf_level	Desired level of confidence for the computed interval (i.e., 1 - the Type I error rate)
R2_Y_X_without_j	The squared multiple correlation coefficient predicting *Y* from the *p*-1 predictor variable with the *j*th predictor of interest excluded
t_value	The *t*-value evaluating the null hypothesis that the population regression coefficient for the *j*th predictor equals zero
alpha_lower	The Type I error rate for the lower confidence interval limit
alpha_upper	The Type I error rate for the upper confidence interval limit
noncentral	TRUE or FALSE statement specifying whether or not the noncentral approach to confidence intervals should be used
...	Optional additional specifications for nested functions

Details

Returns the confidence limits for the regression coefficient of interest from the standard approach to confidence interval formation or from the noncentral approach to confidence interval formation using the noncentral *t*-distribution.

Value

A 2-row data.frame with columns term, value, prob_less, and prob_greater. The term values are "lower_limit" and "upper_limit", and value holds the confidence limits on the regression coefficient in its raw metric. The prob_less and prob_greater columns report the tail probabilities below and above each limit; when the noncentral *t* approach is used they are the achieved tail probabilities. Unlike ci_src and ci_reg_coef, which place the point estimate between its limits as a third row, ci_rc returns the two limits only.

Note

Not all of the values need to be specified, only those that contain all of the necessary information in order to compute the confidence interval (options are thus given for the values that need to be specified).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means and Chapter 6 on trend analysis.)

Smithson, M. (2003). *Confidence intervals*. Thousand Oaks, CA: Sage Publications.

Steiger, J. H. (2004). Beyond the *F* Test: Effect size confidence intervals and tests of close fit in the Analysis of Variance and Contrast Analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

ss_aipe_reg_coef, ci_nc_t, ci_reg_coef, ci_src

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
ci_rc(b_j = 0.61319, SE_b_j = 0.16098, N = 30, p = 6, conf_level = 0.95)
```

---

ci_reg_coef *Confidence Interval for a Regression Coefficient, Raw or Standardized*

Description

The general engine behind ci_rc (unstandardized) and ci_src (standardized): computes a confidence interval for a population regression coefficient by the standard *t*-based approach or the noncentral *t* approach, in whichever metric the inputs are supplied.

Usage

```
ci_reg_coef(
  b_j,
  SE_b_j = NULL,
  s_Y = NULL,
  s_X = NULL,
  N,
  p,
  R2_Y_X = NULL,
  R2_j_X_without_j = NULL,
  conf_level = 0.95,
  R2_Y_X_without_j = NULL,
  t_value = NULL,
  alpha_lower = NULL,
  alpha_upper = NULL,
  noncentral = FALSE,
  ...
)
```

Arguments

b_j	Value of the regression coefficient for the *j*th predictor variable
SE_b_j	Standard error for the *j*th predictor variable
s_Y	Standard deviation of *Y*, the dependent variable
s_X	Standard deviation of X_j, the predictor variable of interest
N	Sample size
p	The number of predictors
R2_Y_X	The squared multiple correlation coefficient predicting Y from the p predictor variables
R2_j_X_without_j	The squared multiple correlation coefficient predicting the jth predictor variable (i.e., the predictor of interest) from the remaining p-1 predictor variables
conf_level	Desired level of confidence for the computed interval (i.e., 1 - the Type I error rate)
R2_Y_X_without_j	The squared multiple correlation coefficient predicting Y from the p-1 predictor variable with the jth predictor of interest excluded
t_value	The *t*-value evaluating the null hypothesis that the population regression coefficient for the jth predictor equals zero
alpha_lower	The Type I error rate for the lower confidence interval limit
alpha_upper	The Type I error rate for the upper confidence interval limit
noncentral	TRUE or FALSE, specifying whether or not the noncentral approach to confidence intervals should be used
...	Optional additional specifications for nested functions

Details

For standardized variables, do not specify the standard deviation of the variables and input the standardized regression coefficient for `b_j`.

When `b_j` is reconstructed from squared multiple correlations (that is, from `R2_Y_X`, `R2_Y_X_without_j`, and `R2_j_X_without_j` rather than a supplied `b_j`, `SE_b_j`, or `t_value`), only the magnitude of the coefficient is identifiable; its sign is not. The positive root is returned and a warning is issued. If the coefficient is negative, negate the point estimate and swap and negate the confidence limits, or supply `b_j` directly.

Value

A 3-row `data.frame` with columns `term`, `value`, `prob_less`, and `prob_greater`. The rows are ordered `"lower_limit"`, `"reg_coef"` (the regression coefficient point estimate), and `"upper_limit"`, so the point estimate sits between its confidence limits. The lower and upper rows give the confidence limits on the regression coefficient. The `prob_less` and `prob_greater` columns report the achieved tail probabilities at each limit when the noncentral t method is used (they are `NA` for the `"reg_coef"` estimate row).

Note

Not all of the values need to be specified, only those that contain all of the necessary information in order to compute the confidence interval (options are thus given for the values that need to be specified).

The function `ci_rc` in DMAR also calculates the confidence interval for the population (unstandardized) regression coefficient. The function `ci_src` also calculates the confidence interval for the population (standardized) regression coefficient. These two functions perform the same tasks as `ci_reg_coef` does and are preferred to it because of simpler arguments.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means and Chapter 6 on trend analysis.)

Smithson, M. (2003). *Confidence intervals*. Thousand Oaks, CA: Sage Publications.

See Also

ss_aipe_reg_coef, ci_nc_t, ci_rc, ci_src

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
ci_reg_coef(b_j = 0.61319, SE_b_j = 0.16098, N = 30, p = 6)
```

---

ci_rmsea *Confidence Interval for the Population Root Mean Square Error of Approximation*

Description

Constructs a confidence interval for the population root mean square error of approximation (RMSEA), a population badness-of-fit index for structural equation models. The interval is obtained by inverting the noncentral chi square distribution of the sample fit function $T = (N-1)\hat{F}_{ML}$ under the model implied covariance structure, mapping the resulting noncentrality limits to the RMSEA metric (Steiger & Lind, 1980; Browne & Cudeck, 1993).

Usage

```
ci_rmsea(
  rmsea,
  df,
  N,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL
)
```

Arguments

rmsea	Observed root mean square error of approximation
df	Degrees of freedom of the model
N	Sample size
conf_level	Desired confidence level (e.g., .90, .95, .99)
alpha_lower	The Type I error rate for the lower tail
alpha_upper	The Type I error rate for the upper tail

Details

The RMSEA expresses the badness of model fit per degree of freedom on the noncentrality scale. Under the noncentral chi square model for the sample fit statistic, the sample $T = (N-1)\hat{F}_{ML}$ has approximate noncentral chi square distribution with df degrees of freedom and noncentrality parameter $\lambda = (N-1)df \cdot \mathrm{RMSEA}^2$. The CI on RMSEA^2 is obtained by inverting the noncentral chi square distribution at the requested confidence level (`ci_nc_chisq` does the inversion); the bounds are then mapped back to the RMSEA scale via the square root. When the lower noncentrality limit hits zero (*i.e.*, the data are compatible with a well-fitting model), the lower RMSEA limit is truncated at zero because RMSEA is non-negative by construction.

The 90 percent CI (rather than the usual 95 percent) is the conventional reporting choice for RMSEA (Browne & Cudeck, 1993) because the upper limit of the 90 percent CI plays a one-sided role in the test of close fit ($H_0 : \mathrm{RMSEA} \leq 0.05$). `ci_rmsea` defaults to `conf_level = 0.95` in line with the rest of the package; pass `conf_level = 0.90` when the close fit test is the intended use.

Value

A 3-row `data.frame` with columns `term` and `value`. The `term` values are `"lower_limit"` (the lower bound of the confidence interval on the population RMSEA, truncated at zero by definition), `"rmsea"` (the observed point estimate), and `"upper_limit"` (the upper bound).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Browne, M. W., & Cudeck, R. (1993). Alternative ways of assessing model fit. In K. A. Bollen & J. S. Long (Eds.), *Testing structural equation models* (pp. 136–162). Sage.

Kelley, K., & Lai, K. (2011). Accuracy in parameter estimation for the root mean square error of approximation: Sample size planning for narrow confidence intervals. *Multivariate Behavioral Research, 46*, 1–32. doi:10.1080/00273171.2011.543027

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Steiger, J. H., & Lind, J. C. (1980). *Statistically-based tests for the number of common factors*. Paper presented at the annual Spring meeting of the Psychometric Society, Iowa City, IA.

See Also

Other confidence intervals for effect sizes: `ci_R2()`, `ci_c()`, `ci_c_ancova()`, `ci_c_ancova_bp()`, `ci_correlation`, `ci_cv()`, `ci_eta_squared()`, `ci_eta_squared_generalized()`, `ci_eta_squared_partial()`, `ci_mahalanobis()`, `ci_omega_squared()`, `ci_pvaf()`, `ci_rc()`, `ci_reg_coef()`, `ci_sc()`, `ci_sc_ancova()`, `ci_sm()`, `ci_smd()`, `ci_smd_c()`, `ci_snr()`, `ci_src()`, `ci_srsnr()`, `contrast_adjusted()`, `plot_smd()`

Examples

```
# 1. A typical 95 percent CI on RMSEA.
ci_rmsea(rmsea = .055, df = 40, N = 425, conf_level = .95)
```

```
# 2. The 90 percent CI is the conventional choice when interpretation
#    will follow the Browne and Cudeck (1993) close fit decision rule
#    (the test of H_0: RMSEA <= 0.05 vs. the upper CI limit). Here the
#    upper limit is 0.052, just above the close fit threshold of 0.05
#    that Browne and Cudeck recommend, so close fit is not established
#    even though the point estimate sits comfortably below it.
ci_rmsea(rmsea = .035, df = 40, N = 425, conf_level = .90)

# 3. Wider model with smaller N: more uncertainty, wider CI.
ci_rmsea(rmsea = .055, df = 10, N = 100, conf_level = .90)
```

`ci_sc` *Confidence Interval for a Standardized Contrast in a Fixed Effects ANOVA*

Description

Computes the exact noncentral *t*-based confidence interval for a standardized contrast of means in a fixed effects analysis of variance: the contrast of interest divided by the error standard deviation, so groups measured in raw units are compared on a common standardized scale.

Usage

```
ci_sc(
  means = NULL,
  s_anova = NULL,
  c_weights = NULL,
  n = NULL,
  N = NULL,
  psi = NULL,
  ncp = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  df_error = NULL,
  ...
)
```

Arguments

`means`	A vector of the group means or the means of the particular level of the effect (for fixed effect designs)
`s_anova`	The standard deviation of the errors from the ANOVA model (i.e., the square root of the mean square error)
`c_weights`	The contrast weights (chose weights so that the positive *c*-weights sum to 1 and the negative *c*-weights sum to -1; i.e., use fractional values not integers).

n	Sample sizes *per group* or sample sizes for the level of the particular factor (if length 1 it is assumed that the sample size *per group* or for the level of the particular factor are are equal)
N	Total sample size
psi	The (unstandardized) contrast effect, obtained by multiplying the *j*th mean by the *j*th contrast weight (this is the unstandardized effect)
ncp	The noncentrality parameter from the *t*-distribution
conf_level	Desired level of confidence for the computed interval (i.e., 1 - the Type I error rate)
alpha_lower	The Type I error rate for the lower confidence interval limit
alpha_upper	The Type I error rate for the upper confidence interval limit
df_error	The degrees of freedom for the error. In one-way designs, this is simply *N*-length (means) and need not be specified; it must be specified if the design has multiple factors.
...	Optional additional specifications for nested functions

Value

A 3-row data.frame with columns term and value. The term values are "lower_limit" (the lower confidence limit on the standardized contrast), "std_contrast" (the standardized contrast), and "upper_limit" (the upper limit).

Note

Be sure to use the standard deviation and not the error variance for s_anova, not the square of this value (the error variance) which would come from the source table (i.e., do not use the variance of the error but rather use its square root, the standard deviation).

Be sure to use fractional *c*-weights when doing complex contrasts (not integers) to specify c_weights. For example, in an ANCOVA of four groups, if the user wants to compare the mean of group 1 and 2 with the mean of group 3 and 4, c_weights should be specified as c(0.5, 0.5, -0.5, -0.5) rather than c(1, 1, -1, -1). Make sure the sum of the contrast weights are zero.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Steiger, J. H. (2004). Beyond the *F* Test: Effect size confidence intervals and tests of close fit in the Analysis of Variance and Contrast Analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

ci_nc_t, ci_src, ci_smd, ci_smd_c, ci_sm, ci_c ci_c_ancova

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
# Here is a four group example. Suppose that the means of groups 1--4 are 2, 4, 9,
# and 13, respectively. Further, let the error variance be .64 and thus the standard
# deviation would be .80 (note we use the standard deviation in the function, not the
# variance). The standardized contrast of interest here is the average of groups 1 and 4
# versus the average of groups 2 and 3.

ci_sc(means = c(2, 4, 9, 13), s_anova = .80, c_weights = c(.5, -.5, -.5, .5),
      n = c(3, 3, 3, 3), N = 12, conf_level = .95)

# Here is an example with two groups.
ci_sc(means = c(1.6, 0), s_anova = .80, c_weights = c(1, -1),
      n = c(10, 10), N = 20, conf_level = .95)
```

---

ci_sc_ancova *Confidence Interval for a Standardized Contrast in ANCOVA With One Covariate*

Description

Calculate the confidence interval for a standardized contrast in ANCOVA with one covariate. The standardizer (i.e., the divisor) can be either the error standard deviation of the ANOVA model (i.e., the model excluding the covariate) or of the ANCOVA model.

Usage

```
ci_sc_ancova(
  psi = NULL,
  adj_means = NULL,
  s_anova = NULL,
  s_ancova = NULL,
  standardizer = "s_ancova",
  c_weights,
  n,
  cov_means,
  SSwithin_x,
  conf_level = 0.95
)
```

Arguments

psi	Unstandardized contrast of adjusted means
adj_means	The vector that contains the adjusted mean of each group on the dependent variable
s_anova	The standard deviation of the errors from the ANOVA model (i.e., the square root of the mean square error from ANOVA)
s_ancova	The standard deviation of the errors from the ANCOVA model (i.e., the square root of the mean square error from ANCOVA)
standardizer	Which error standard deviation the user wants to use, the value of which can be either "s_ancova" or "s_anova"
c_weights	The contrast weights (chose weights so that the positive *c*-weights sum to 1 and the negative *c*-weights sum to -1; i.e., use fractional values not integers).
n	Either a single number that indicates the sample size per group, or a vector that contains the sample size of each group
cov_means	A vector that contains the group means of the covariate
SSwithin_x	The sum of squares within groups obtained from the summary table for ANOVA on the covariate
conf_level	The desired confidence interval coverage, (i.e., 1 - Type I error rate)

Details

The argument SSwithin_x is the sum of squares within groups for the covariate, taken from the ANOVA source table in which the covariate (not the outcome) is the dependent variable. Published reports do not always print this quantity directly. When a report gives the covariate group means, the group sample sizes, and the *F* statistic from the one-way ANOVA on the covariate, SSwithin_x can be recovered algebraically. The worked example below follows Lai and Kelley (2012): three groups of sizes 19, 18, and 19 (so $N = 56$) have covariate means 60.08, 57.08, and 57.97, and the covariate ANOVA reports $F = 0.756$ with 2 and 53 degrees of freedom. The sum of squares between groups for the covariate, computed from the group means and sample sizes, is approximately 88.5, so the mean square between groups is approximately $88.5/2 = 44.3$. Because *F* is the ratio of the mean square between groups to the mean square within groups, the mean square within groups is approximately $44.3/0.756 = 58.6$, and the sum of squares within groups is that mean square times its degrees of freedom, approximately $58.6 \times 53 = 3103$. That recovered value is what you would pass to SSwithin_x. The "Examples" section reproduces this computation in code.

Value

A 3-row data.frame with columns term and value (numeric). The term values are "lower_limit" (the lower confidence limit on the standardized ANCOVA contrast), "psi" (the standardized contrast), and "upper_limit" (the upper limit). The divisor used in standardization (either "s_anova" or "s_ancova") is attached as the "standardizer" attribute of the returned data.frame.

Note

Be sure to use the standard deviations and not the error variances for s_anova and s_ancova, not the squares of these values which would come from the source tables (i.e., do not use the variance of the errors but rather use its square root, the standard deviation).

If n receives a single number, that number is considered as the sample size per group. If n is assigned to a vector, the vector is considered as the sample size of each group.

Be sure to use fractional *c*-weights when doing complex contrasts (not integers) to specify c_weights. For example, in an ANCOVA of four groups, if the user wants to compare the mean of group 1 and 2 with the mean of group 3 and 4, c_weights should be specified as c(0.5, 0.5, -0.5, -0.5) rather than c(1, 1, -1, -1). Make sure the sum of the contrast weights are zero.

The argument to be assigned to standardizer must be either "s_ancova" or "s_anova".

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*, 363–385. doi:10.1037/1082989X.11.4.363

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9.)

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

ci_c_ancova, ci_sc

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
# Maxwell, Delaney, & Kelley (2027) offer an example that 30 depressive
# individuals are randomly assigned to three groups, 10 in each, and ANCOVA
# is performed on the posttest scores using the participants' pretest
# scores as the covariate. The means of pretest scores of group 1, 2, and 3 are
# 17, 17.7, and 17.4, respectively, whereas the adjusted means of groups 1, 2, and 3
# are 7.5, 12, and 14, respectively. The error variance in ANCOVA is 29 and thus
# 5.385165 is the error standard deviation, with the sum of squares within groups
# from an ANOVA on the covariate is 752.5.
```

```
# To obtained the confidence interval for the standardized adjusted mean difference
# between group 1 and 2, using the ANCOVA error standard deviation:
ci_sc_ancova(adj_means = c(7.5, 12, 14), s_ancova = 5.385165, c_weights = c(1, -1, 0),
             n = 10, cov_means = c(17, 17.7, 17.4), SSwithin_x = 752.5)

# Or, with less error in rounding:
ci_sc_ancova(adj_means = c(7.54, 11.98, 13.98), s_ancova = 5.393, c_weights = c(-1, 0, 1),
             n = 10, cov_means = c(17, 17.7, 17.4), SSwithin_x = 752.5)

# Now, using the standard deviation from ANOVA (and not ANCOVA as above), we have:
ci_sc_ancova(adj_means = c(7.54, 11.98, 13.98), s_anova = 6.294, s_ancova = 5.393,
             c_weights = c(-1, 0, 1),n = 10, cov_means = c(17, 17.7, 17.4),
             SSwithin_x = 752.5, standardizer = "s_anova", conf_level = .95)

# Recovering SSwithin_x from a covariate ANOVA F when a report does not print
# it directly (see the Details section). This example follows Lai and Kelley
# (2012): three groups of sizes 19, 18, and 19 have covariate means 60.08,
# 57.08, and 57.97, and the one-way ANOVA on the covariate reports F = 0.756.
cov_means_ex  <- c(60.08, 57.08, 57.97)
n_ex          <- c(19, 18, 19)
grand_x       <- sum(n_ex * cov_means_ex) / sum(n_ex)
ss_between_x  <- sum(n_ex * (cov_means_ex - grand_x)^2)
ms_between_x  <- ss_between_x / (length(n_ex) - 1)
ms_within_x   <- ms_between_x / 0.756
SSwithin_x_ex <- ms_within_x * (sum(n_ex) - length(n_ex))

ci_sc_ancova(adj_means = c(63.88, 62.39, 56.48), s_ancova = 20.267,
             c_weights = c(0.5, 0.5, -1), n = n_ex, cov_means = cov_means_ex,
             SSwithin_x = SSwithin_x_ex)
```

`ci_scheffe` *Scheffe-Adjusted Simultaneous Confidence Intervals for Contrasts*

Description

Computes the Scheffe (1953, 1959) simultaneous confidence intervals on user-specified contrasts among the means of a one-way design. The Scheffe procedure controls the family-wise error rate for *any* set of contrasts, however many and however post-hoc, which makes it more conservative than Tukey-Kramer or Bonferroni for the specific case of all-pairwise comparisons but optimal for arbitrary post-hoc contrasts.

Usage

```
ci_scheffe(x, group = NULL, contrasts = NULL, conf_level = 0.95)
```

Arguments

`x`	A fitted `lm` or `aov` object with a single one-way factor predictor, or a numeric vector of observations with `group` supplied.

group	Optional factor of group labels when x is a numeric vector.
contrasts	An $a \times m$ matrix or vector of contrast coefficients (rows = levels, columns = contrasts). Each column must sum to zero. If NULL (default), the function returns intervals for all $a(a-1)/2$ pairwise contrasts.
conf_level	Family-wise confidence level. Default 0.95.

Details

Critical value. For a groups with ν error degrees of freedom, the Scheffe critical value is

$$S \;=\; \sqrt{(a-1)F_{1-\alpha,a-1,\nu}},$$

where $F_{1-\alpha,a-1,\nu}$ is the upper α quantile of the central F distribution. The Scheffe simultaneous CI on a contrast $\psi = \sum_i c_i \mu_i$ is

$$\hat{\psi} \;\pm\; S \cdot SE(\hat{\psi}),$$

where $SE(\hat{\psi}) = \sqrt{MS_E \sum_i c_i^2 / n_i}$.

Scope. The Scheffe family-wise coverage holds for *any* number of contrasts, pairwise, complex, or chosen after looking at the data. The trade-off is conservativeness: for all-pairwise comparisons, Tukey-Kramer is uniformly more powerful.

Value

A data.frame with one row per contrast. Columns: contrast (a printed label), contrast_value, se, F_statistic, lower_limit, upper_limit, p_adjusted.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Scheffe, H. (1953). A method for judging all contrasts in the analysis of variance. *Biometrika, 40*(1/2), 87–104.

Scheffe, H. (1959). *The analysis of variance*. Wiley.

See Also

cv_scheffe, ci_tukey_kramer, ci_dunnett

Other hypothesis tests: adjusted_means(), ancova(), anova_within(), ci_dunnett(), ci_tukey_kramer(), compare_cov_structures(), contrast_test(), correlations_test(), equivalence_r(), equivalence_smd(), factorial_anova(), manova_split_plot(), mauchly_test(), mixed_anova(), obrien_test(), pairwise_within(), randomization_test(), randomization_test_paired(), regions_of_significance(), simple_effects_AB(), summary_t_test(), welch_t()

Examples

```
# 1. All pairwise contrasts among the six marketing panels of the
#    test_market data via the default:
fit <- lm(brand_movement ~ panel, data = test_market)
ci_scheffe(fit)

# 2. A contrast chosen after inspecting the means: the two panels with
#    the highest brand movement (5 and 6) against the two with the
#    lowest (1 and 2). The Scheffe coverage holds for a contrast picked
#    this way, and the interval still excludes zero even though none of
#    the pairwise intervals above does.
cmat <- matrix(c(-0.5, -0.5, 0, 0, 0.5, 0.5), nrow = 6,
               dimnames = list(levels(test_market$panel),
                               "panels 5,6 - panels 1,2"))
ci_scheffe(fit, contrasts = cmat)
```

`ci_sm` *Confidence Interval for the Standardized Mean*

Description

Computes the exact confidence interval for the standardized mean, the mean divided by the standard deviation, by inverting the noncentral t distribution. The standardized mean is the one-sample analog of the standardized mean difference and shares its noncentral interval theory.

Usage

```
ci_sm(
  sm = NULL,
  mean = NULL,
  sd = NULL,
  ncp = NULL,
  N = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  ...
)
```

Arguments

`sm`	Standardized mean
`mean`	Mean
`sd`	Standard deviation
`ncp`	Noncentral parameter
`N`	Sample size

conf_level	Confidence interval coverage (i.e., 1 - Type I error rate); default is .95
alpha_lower	Type I error for the lower confidence limit
alpha_upper	Type I error for the upper confidence limit
...	Allows one to potentially include parameter values for inner functions

Details

The user must specify the standardized mean in one and only one of the three ways: a) mean and standard deviation (mean and sd), b) standardized mean (sm), and c) noncentral parameter (ncp). The confidence level must be specified in one of following two ways: using confidence interval coverage (conf_level), or lower and upper confidence limits (alpha_lower and alpha_upper). This function uses the exact confidence interval method based on noncentral *t*-distributions. The confidence interval for noncentral *t*-parameter can be obtained from the ci_nc_t function in DMAR.

Value

A 3-row data.frame with columns term and value. The term values are "lower_limit" (the lower confidence limit on the standardized mean), "std_mean" (the standardized mean), and "upper_limit" (the upper confidence limit).

Note

The standardized mean is the mean divided by the standard deviation.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

ci_nc_t

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
ci_sm(sm = 2.037905, N = 13, conf_level = .95)
ci_sm(mean = 30, sd = 14.721, N = 13, conf_level = .95)
ci_sm(ncp = 7.347771, N = 13, conf_level = .95)
ci_sm(sm = 2.037905, N = 13, alpha_lower = .05, alpha_upper = 0)
ci_sm(mean = 50, sd = 10, N = 25, conf_level = .95)
```

---

ci_smd *Confidence Interval for the Standardized Mean Difference (Two Independent Groups)*

Description

Constructs an exact-coverage confidence interval for the population standardized mean difference $\delta = (\mu_1 - \mu_2)/\sigma$ (Cohen's d when expressed as a sample quantity) for two independent groups under bivariate normality with equal variances. The interval is obtained by inverting the noncentral t sampling distribution of the rescaled statistic $t = \hat{d}\sqrt{n_1 n_2/(n_1 + n_2)}$, which under the independent groups equal variances model is exactly noncentral t with $n_1 + n_2 - 2$ degrees of freedom and noncentrality parameter $\lambda = \delta\sqrt{n_1 n_2/(n_1 + n_2)}$ (Hedges, 1981). The confidence limits are then rescaled back to the δ metric. This is the same construction Steiger and Fouladi (1997) and Kelley (2007) describe for noncentral effect size CIs.

Usage

```
ci_smd(
  ncp = NULL,
  smd = NULL,
  n_1 = NULL,
  n_2 = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  tol = 1e-09,
  ...
)
```

Arguments

ncp	The estimated noncentrality parameter, this is generally the observed *t*-statistic from comparing the two groups and assumes homogeneity of variance
smd	The standardized mean difference (using the pooled standard deviation in the denominator)
n_1	The sample size for Group 1
n_2	The sample size for Group 2
conf_level	The confidence level (1-Type I error rate)

alpha_lower	The Type I error rate for the lower tail
alpha_upper	The Type I error rate for the upper tail
tol	The tolerance of the iterative method for determining the critical values
...	Allows one to potentially include parameter values for inner functions

Details

ncp-input vs. smd-input paths. The function accepts the effect size in either of two equivalent metrics: the observed *t*-statistic (via ncp) or the sample standardized mean difference (via smd). The two paths are mathematically equivalent under the equal variances assumption (since $t = \hat{d}\sqrt{n_1 n_2/(n_1+n_2)}$); pick whichever is easier to obtain. Supply exactly one. Both paths internally call ci_nc_t to invert the noncentral *t* distribution at the specified two-tailed (or asymmetric, via alpha_lower / alpha_upper) confidence level.

Independent vs.\ paired comparison. ci_smd assumes two *independent* groups with a common variance. DMAR does not currently provide a confidence interval for the standardized mean difference in a paired or within-subject design, whose sampling distribution depends on the correlation between the paired measurements; applying the independent groups interval to paired data gives the wrong coverage. (ci_smd_c is not a paired interval either; it is the interval for Glass's estimator, which standardizes the difference between two independent groups by the control group standard deviation.)

Bias correction (Hedges' g). ci_smd reports the CI on *d*; if the bias-corrected *g* is desired, multiply the bounds by the Hedges and Olkin (1985) correction factor $J(\nu) = 1 - 3/(4\nu - 1)$ (with $\nu = n_1 + n_2 - 2$). Because $J(\nu)$ is a constant, the rescaling preserves coverage.

Value

A 3-row data.frame with columns term and value. The term values are "lower_limit" (the lower bound of the confidence interval on the standardized mean difference), "smd" (the point estimate), and "upper_limit" (the upper bound).

Warning

This function uses ci_nc_t, which has as one of its arguments tol (and can be modified with tol of the present function). If the present function fails to converge (i.e., if it runs but does not report a solution), it is likely that the tol value is too restrictive and should be increased by a factor of 10, but probably by no more than 100. Running the function ci_nc_t directly will report the actual probability values of the limits found. This should be done if any modification to tol is necessary in order to ensure acceptable confidence limits for the noncentral *t* parameter have been achieved.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*(4), 532–574. doi:10.1177/0013164401614002

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. Academic Press.

Kelley, K. (2005). The effects of nonnormal distributions on confidence intervals around the standardized mean difference: Bootstrap and parametric confidence intervals. *Educational and Psychological Measurement, 65*(1), 51–69. doi:10.1177/0013164404264850

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Kelley, K., Maxwell, S. E., & Rausch, J. R. (2003). Obtaining power or obtaining precision: Delineating methods of sample size planning. *Evaluation and the Health Professions, 26*(3), 258–287. doi:10.1177/0163278703255242

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons and Chapter 3 on one-way ANOVA.)

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

`smd`, `smd_c`, `ci_smd_c`, `ss_aipe_smd`, `ss_power_smd`, `plot_smd`, `ci_nc_t`

Other confidence intervals for effect sizes: `ci_R2()`, `ci_c()`, `ci_c_ancova()`, `ci_c_ancova_bp()`, `ci_correlation`, `ci_cv()`, `ci_eta_squared()`, `ci_eta_squared_generalized()`, `ci_eta_squared_partial()`, `ci_mahalanobis()`, `ci_omega_squared()`, `ci_pvaf()`, `ci_rc()`, `ci_reg_coef()`, `ci_rmsea()`, `ci_sc()`, `ci_sc_ancova()`, `ci_sm()`, `ci_smd_c()`, `ci_snr()`, `ci_src()`, `ci_srsnr()`, `contrast_adjusted()`, `plot_smd()`

Examples

```
# Steiger and Fouladi (1997) example values.
ci_smd(ncp = 2.6, n_1 = 10, n_2 = 10, conf_level = 1 - .05)
ci_smd(ncp = 2.4, n_1 = 300, n_2 = 300, conf_level = 1 - .05)
```

`ci_smd_c` *Confidence Limits for the Standardized Mean Difference Using the Control Group Standard Deviation as the Divisor*

Description

Computes the exact noncentral *t*-based confidence limits for the standardized mean difference that uses the control group standard deviation as the divisor (Glass's *g*). Standardizing by the control group alone keeps the scale of the effect anchored in the untreated population, which matters when the treatment may alter variability as well as the mean.

Usage

```
ci_smd_c(
  ncp = NULL,
  smd_c = NULL,
  n_C = NULL,
  n_E = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  tol = 1e-09,
  ...
)
```

Arguments

`ncp`	The estimated noncentrality parameter, this is generally the observed *t*-statistic from comparing the control and experimental group (assuming homogeneity of variance)
`smd_c`	The standardized mean difference (using the control group standard deviation in the denominator)
`n_C`	The sample size for the control group
`n_E`	The sample size for experimental group
`conf_level`	The confidence level (1-Type I error rate)
`alpha_lower`	The Type I error rate for the lower tail
`alpha_upper`	The Type I error rate for the upper tail
`tol`	The tolerance of the iterative method for determining the critical values
`...`	Potentially include parameter for inner functions

Value

A 3-row `data.frame` with columns `term` and `value`. The `term` values are `"lower_limit"` (the lower bound of the confidence interval), `"smd_c"` (the standardized mean difference standardized by the control group standard deviation), and `"upper_limit"` (the upper bound).

Warning

This function uses `ci_nc_t`, which has as one of its arguments `tol` (and can be modified with `tol` of the present function). If the present function fails to converge (i.e., if it runs but does not report a solution), it is likely that the `tol` value is too restrictive and should be increased by a factor of 10, but probably by no more than 100. Running the function `ci_nc_t` directly will report the actual probability values of the limits found. This should be done if any modification to `tol` is necessary in order to ensure acceptable confidence limits for the noncentral *t* parameter have been achieved.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*(4), 532–574. doi:10.1177/0013164401614002

Glass, G. V. (1976). Primary, secondary, and meta-analysis of research. *Educational Researcher, 5*, 3–8.

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons and Chapter 3 on one-way ANOVA.)

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

`smd_c`, `smd`, `ci_smd`, `ci_nc_t`

Other confidence intervals for effect sizes: `ci_R2()`, `ci_c()`, `ci_c_ancova()`, `ci_c_ancova_bp()`, `ci_correlation`, `ci_cv()`, `ci_eta_squared()`, `ci_eta_squared_generalized()`, `ci_eta_squared_partial()`, `ci_mahalanobis()`, `ci_omega_squared()`, `ci_pvaf()`, `ci_rc()`, `ci_reg_coef()`, `ci_rmsea()`, `ci_sc()`, `ci_sc_ancova()`, `ci_sm()`, `ci_smd()`, `ci_snr()`, `ci_src()`, `ci_srsnr()`, `contrast_adjusted()`, `plot_smd()`

Examples

```
ci_smd_c(smd_c = .5, n_C = 100, n_E = 100, conf_level = .95)
```

`ci_snr` *Confidence Interval for the Signal-to-Noise Ratio*

Description

Computes the exact confidence interval for the signal-to-noise ratio in a fixed effects analysis of variance, the variance due to the factor of interest divided by the error variance, expressing the magnitude of an effect relative to the unexplained variability.

Usage

```
ci_snr(
  F_value = NULL,
  df_1 = NULL,
  df_2 = NULL,
  N = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  ...
)
```

Arguments

`F_value`	Observed *F*-value from the analysis of variance
`df_1`	Numerator degrees of freedom
`df_2`	Denominator degrees of freedom
`N`	Sample size
`conf_level`	Confidence interval coverage (i.e., 1 - Type I error rate), default is .95
`alpha_lower`	Type I error for the lower confidence limit
`alpha_upper`	Type I error for the upper confidence limit
`...`	Allows one to potentially include parameter values for inner functions

Details

The confidence level must be specified in one of following two ways: using confidence interval coverage (`conf_level`), or lower and upper confidence limits (`alpha_lower` and `alpha_upper`).

This function uses the confidence interval transformation principle (Steiger, 2004) to transform the confidence limits for the noncentrality parameter to the confidence limits for the population's signal-to-noise ratio. The confidence interval for noncentral *F* parameter can be obtained from the `ci_nc_F` function in DMAR, which is used internally within this function.

Value

A 2-row `data.frame` with columns `term` and `value`. The `term` values are `"lower_limit"` and `"upper_limit"`, giving the lower and upper confidence limits on the signal-to-noise ratio.

Note

The signal to noise ratio is defined as the variance due to the particular factor over the error variance (i.e., the mean square error).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Fleishman, A. I. (1980). Confidence intervals for correlation ratios. *Educational and Psychological Measurement, 40*(3), 659–670.

Steiger, J. H. (2004). Beyond the *F* Test: Effect size confidence intervals and tests of close fit in the Analysis of Variance and Contrast Analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

`ci_srsnr`, `ci_nc_F`

Other confidence intervals for effect sizes: `ci_R2()`, `ci_c()`, `ci_c_ancova()`, `ci_c_ancova_bp()`, `ci_correlation`, `ci_cv()`, `ci_eta_squared()`, `ci_eta_squared_generalized()`, `ci_eta_squared_partial()`, `ci_mahalanobis()`, `ci_omega_squared()`, `ci_pvaf()`, `ci_rc()`, `ci_reg_coef()`, `ci_rmsea()`, `ci_sc()`, `ci_sc_ancova()`, `ci_sm()`, `ci_smd()`, `ci_smd_c()`, `ci_src()`, `ci_srsnr()`, `contrast_adjusted()`, `plot_smd()`

Examples

```
## Bargman (1970) gave an example in which a 5-group ANOVA with 11 subjects in each
## group is conducted and the observed F value is 11.221. This example was
## used in Venables (1975), Fleishman (1980), and Steiger (2004). If one wants to calculate
## the exact confidence interval for the signal-to-noise ratio of that example, this
## function can be used.

ci_snr(F_value = 11.221, df_1 = 4, df_2 = 50, N = 55)

ci_snr(F_value = 11.221, df_1 = 4, df_2 = 50, N = 55, conf_level = .90)

ci_snr(F_value = 11.221, df_1 = 4, df_2 = 50, N = 55, alpha_lower = .02, alpha_upper = .03)
```

ci_src *Confidence Interval for a Standardized Regression Coefficient*

Description

Computes a confidence interval for a population standardized regression coefficient, by the standard *t*-based approach or the noncentral *t* approach. A thin convenience wrapper around ci_reg_coef, which is the general engine; for the coefficient in its raw metric use ci_rc.

Usage

```
ci_src(
  beta_j = NULL,
  SE_beta_j = NULL,
  N = NULL,
  p = NULL,
  R2_Y_X = NULL,
  R2_j_X_without_j = NULL,
  conf_level = 0.95,
  R2_Y_X_without_j = NULL,
  t_value = NULL,
  b_j = NULL,
  SE_b_j = NULL,
  s_Y = NULL,
  s_X = NULL,
  alpha_lower = NULL,
  alpha_upper = NULL,
  ...
)
```

Arguments

beta_j	The standardized regression coefficient
SE_beta_j	The standard error of the standardized regression coefficient
N	Sample size
p	The number of predictors
R2_Y_X	The squared multiple correlation coefficient predicting *Y* from the *p* predictor variables
R2_j_X_without_j	The squared multiple correlation coefficient predicting the *j*th predictor variable (i.e., the predictor of interest) from the remaining *p*-1 predictor variables
conf_level	Desired level of confidence for the computed interval (i.e., 1 - the Type I error rate)
R2_Y_X_without_j	The squared multiple correlation coefficient predicting *Y* from the *p*-1 predictor variable with the *j*th predictor of interest excluded

t_value	The *t*-value evaluating the null hypothesis that the population regression coefficient for the *j*th predictor equals zero
b_j	The unstandardized regression coefficient
SE_b_j	The standard error of the unstandardized regression coefficient
s_Y	Standard deviation of *Y*, the dependent variable
s_X	Standard deviation of *X*, the predictor variable of interest
alpha_lower	The Type I error rate for the lower confidence interval limit
alpha_upper	The Type I error rate for the upper confidence interval limit
...	Optional additional specifications for nested functions

Details

For standardized variables, do not specify the standard deviation of the variables and input the standardized regression coefficient for b_j.

Value

A 3-row data.frame with columns term, value, prob_less, and prob_greater. The term values are "lower_limit", "src" (the standardized regression coefficient point estimate), and "upper_limit", so the estimate sits between its confidence limits. The prob_less and prob_greater columns report the achieved tail probabilities at each limit when the noncentral t method is used (NA for the estimate row).

Note

This function calls upon ci_reg_coef in DMAR, but has a different naming scheme. See ci_reg_coef for more details.

To form a confidence interval for the unstandardized regression coefficient, use ci_rc. This function is used to form a confidence interval for the standardized regression coefficient.

Not all of the values need to be specified, only those that contain all of the necessary information in order to compute the confidence interval (options are thus given for the values that need to be specified).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means and Chapter 6 on trend analysis.)

Smithson, M. (2003). *Confidence intervals*. Thousand Oaks, CA: Sage Publications.

Steiger, J. H. (2004). Beyond the *F* Test: Effect size confidence intervals and tests of close fit in the Analysis of Variance and Contrast Analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

ss_aipe_reg_coef, ci_nc_t, ci_reg_coef, ci_rc

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_srsnr(), contrast_adjusted(), plot_smd()

Examples

```
ci_src(beta_j = .6707, .1761, N = 30, p = 6, conf_level = .95)
```

---

ci_srsnr *Confidence Interval for the Square Root of the Signal-to-Noise Ratio*

Description

Computes the exact confidence interval for the square root of the signal-to-noise ratio, the standard deviation of the group means relative to the error standard deviation. On this root scale the quantity is an effect size in standard deviation units, the multi-group analog of the standardized mean difference.

Usage

```
ci_srsnr(
  F_value = NULL,
  df_1 = NULL,
  df_2 = NULL,
  N = NULL,
  means = NULL,
  sigma_squared = NULL,
  n_per_group = NULL,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  ...
)
```

Arguments

F_value	Observed *F*-value from the analysis of variance. Use this argument when re-analyzing existing data.
df_1	Numerator degrees of freedom
df_2	Denominator degrees of freedom
N	Sample size
means	Numeric vector of population or hypothesized group means. Supply together with sigma_squared and n_per_group as a design-stage alternative to F_value: the function then computes the *F*-value implied by these design parameters and proceeds with the same noncentral F machinery.
sigma_squared	The within-group variance. Used with means.
n_per_group	A single per-group sample size, or a vector of per-group sample sizes the same length as means. Used with means.
conf_level	Confidence interval coverage (i.e., 1 - Type I error rate); default is .95
alpha_lower	Type I error for the lower confidence limit
alpha_upper	Type I error for the upper confidence limit
...	Allows one to potentially include parameter values for inner functions

Details

The confidence level must be specified in one of following two ways: using confidence interval coverage (conf_level), or lower and upper confidence limits (alpha_lower and alpha_upper).

The square root of the signal-to-noise ratio is defined as the standard deviation due to the particular factor over the standard deviation of the error (i.e., the square root of the mean square error). This function uses the confidence interval transformation principle (Steiger, 2004) to transform the confidence limits for the noncentrality parameter to the confidence limits for square root of signal-to-noise ratio. The confidence interval for noncentral *F* parameter can be obtained from function ci_nc_F in DMAR.

Value

A 2-row data.frame with columns term and value. The term values are "lower_limit" and "upper_limit", giving the square roots of the corresponding signal-to-noise-ratio confidence limits.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Fleishman, A. I. (1980). Confidence intervals for correlation ratios. *Educational and Psychological Measurement, 40*(3), 659–670.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Steiger, J. H. (2004). Beyond the *F* Test: Effect size confidence intervals and tests of close fit in the Analysis of Variance and Contrast Analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

ci_snr, ci_nc_F

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), contrast_adjusted(), plot_smd()

Examples

```
## To illustrate the calculation of the confidence interval for noncentral
## F parameter,Bargman (1970) gave an example in which a 5-group ANOVA with
## 11 subjects in each group is conducted and the observed F value is 11.221.
## This example continued to be used in Venables (1975),  Fleishman (1980),
## and Steiger (2004). If one wants to calculate the exact confidence interval
## for square root of the signal-to-noise ratio of that example, this
## function can be used.

ci_srsnr(F_value = 11.221, df_1 = 4, df_2 = 50, N = 55)

ci_srsnr(F_value = 11.221, df_1 = 4, df_2 = 50, N = 55, conf_level = .90)

ci_srsnr(F_value = 11.221, df_1 = 4, df_2 = 50, N = 55, alpha_lower = .02, alpha_upper = .03)

# Design-stage call with population means + within-group variance + n.
# Useful when planning a study, before data are observed: derives the
# implied F-value internally and returns the resulting CI on the square
# root of the signal-to-noise ratio.
ci_srsnr(means = c(94, 91, 92, 83), sigma_squared = 67.375, n_per_group = 6)
```

---

ci_tukey_kramer *Tukey-Kramer Simultaneous Confidence Intervals for Pairwise Contrasts*

Description

Computes the Tukey-Kramer simultaneous confidence intervals for all $a(a-1)/2$ pairwise contrasts among a group means in a one-way design with possibly unequal group sample sizes (Tukey, 1953; Kramer, 1956; Hayter, 1984), and returns the result in tidy long form. Each interval has individual coverage at least the specified `conf_level` and the family-wise coverage is at least `conf_level`.

Usage

```
ci_tukey_kramer(x, group = NULL, conf_level = 0.95)
```

Arguments

x	Either (a) a fitted lm or aov object with a single one-way factor predictor, or (b) a numeric vector of observations, in which case group must also be supplied.
group	When x is a vector, a factor (or coercible to factor) of group labels, same length as x.
conf_level	Family-wise confidence level. Default 0.95.

Details

Formula. For groups i, j with means $\bar{y}_i, \bar{y}_j$ and sample sizes n_i, n_j, the Tukey-Kramer simultaneous CI is

$$\bar{y}_i - \bar{y}_j \;\pm\; q_{\alpha,a,\nu}\sqrt{\frac{MS_E}{2}\left(\frac{1}{n_i}+\frac{1}{n_j}\right)},$$

where $q_{\alpha,a,\nu}$ is the upper α quantile of the studentized-range distribution with a groups and ν error degrees of freedom (stats::qtukey()).

Why Tukey-Kramer. Hayter (1984) proved that the Tukey-Kramer procedure is conservative for unbalanced designs (the coverage probability is at least conf_level). For balanced designs it reduces to Tukey's HSD and the coverage is exactly conf_level.

Adjusted *p*-values. Each pairwise p-value is computed from the studentized-range distribution: $p = 1 - \mathrm{ptukey}(|q|, a, \nu)$.

Value

A data.frame with one row per pairwise contrast. Columns: contrast (e.g., "B - A"), mean_difference, se, q_statistic (the studentized-range q), lower_limit, upper_limit, p_adjusted.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Hayter, A. J. (1984). A proof of the conjecture that the Tukey-Kramer multiple comparisons procedure is conservative. *Annals of Statistics, 12*(1), 61–75.

Kramer, C. Y. (1956). Extension of multiple range tests to group means with unequal numbers of replications. *Biometrics, 12*(3), 307–310.

Tukey, J. W. (1953). *The problem of multiple comparisons*. Unpublished manuscript, Princeton University.

See Also

cv_tukey_hsd, ci_dunnett, ci_scheffe, TukeyHSD

Other hypothesis tests: adjusted_means(), ancova(), anova_within(), ci_dunnett(), ci_scheffe(), compare_cov_structures(), contrast_test(), correlations_test(), equivalence_r(), equivalence_smd(), factorial_anova(), manova_split_plot(), mauchly_test(), mixed_anova(), obrien_test(), pairwise_within(), randomization_test(), randomization_test_paired(), regions_of_significance(), simple_effects_AB(), summary_t_test(), welch_t()

Examples

```
# 1. Balanced one-way: the six marketing panels of the test_market
#    data, four outlets per panel, so the procedure is exactly Tukey's
#    HSD. Panels 5 and 6 separate from panel 1.
fit <- lm(brand_movement ~ panel, data = test_market)
ci_tukey_kramer(fit)

# 2. Same data via vector / group interface:
ci_tukey_kramer(test_market$brand_movement, group = test_market$panel)
```

`cles` *Common-Language Effect Size (McGraw & Wong, 1992)*

Description

Computes the common-language (CL) effect size for two independent groups, defined as the probability that a randomly drawn observation from group 1 exceeds a randomly drawn observation from group 2 under bivariate normality with equal variances:

$$\mathrm{CL} \;=\; \Pr(Y_1 > Y_2) \;=\; \Phi\big(\delta/\sqrt{2}\big),$$

where δ is the population standardized mean difference and Φ is the standard normal cumulative distribution function. When sample sizes are supplied, the confidence interval on CL is constructed by transforming the noncentral *t*-based CI on Cohen's *d* (Steiger & Fouladi, 1997; Kelley, 2007) through $\Phi(\cdot/\sqrt{2})$, which is monotone-increasing so the coverage probability is preserved exactly. This is preferred over the normal-approximation CI on CL commonly seen in applied work (Brooks, Dalal, & Nolan, 2014).

Usage

```
cles(
  smd,
  n_1 = NULL,
  n_2 = NULL,
  conf_level = 0.95,
  smd_lower = NULL,
  smd_upper = NULL
)
```

Arguments

`smd`	Sample standardized mean difference (Cohen's *d*); a numeric scalar. Positive means group 1 exceeds group 2.
`n_1`, `n_2`	Sample sizes in the two groups; both required when a confidence interval on CL is desired.
`conf_level`	Confidence level for the CI. Default `0.95`.
`smd_lower`, `smd_upper`	Optional pre-computed confidence limits on *d*. If supplied, these are used directly and the noncentral computation is skipped.

Details

The common-language idea extends to other effect sizes; the common language effect size for correlations is developed by Liu, Carlson, and Kelley (2019).

Background. McGraw & Wong (1992) introduced the CL effect size to make Cohen's d more interpretable: instead of "the means differ by 0.5 SD," one can say "in 64 treated person scores higher than the control person." Under bivariate normality with equal variances, the population probability $\Pr(Y_1 > Y_2)$ equals $\Phi(\delta/\sqrt{2})$, where $\delta = (\mu_1 - \mu_2)/\sigma$ (McGraw & Wong, 1992).

Connection to other measures. CL is identical to the AUC (Area Under the Curve) interpretation of d in receiver-operating analysis. Vargha & Delaney (2000) generalized CL to the nonparametric setting (their A measure) by replacing the population p with its empirical Mann-Whitney estimate; under bivariate normality the two coincide. The success-rate-difference and number-needed-to-treat scales (Kraemer & Kupfer, 2006; see `nnt_from_smd`) are linear transformations of CL: $\mathrm{SRD} = 2\mathrm{CL} - 1$.

Confidence interval construction. Because $\Phi(\cdot/\sqrt{2})$ is monotone-increasing, the CI on CL is obtained by transforming the CI on d: $[\Phi(d_L/\sqrt{2}),\ \Phi(d_U/\sqrt{2})]$. This is an exact-coverage interval (under the noncentral t sampling model) and is more accurate than the normal-approximation CI on CL that uses a Wald-style variance for $\hat{p}$ (Brooks, Dalal, & Nolan, 2014).

Value

A `data.frame` with rows for the point estimate (`cl`) and, when sample sizes are supplied, the lower and upper CI limits. The d-equivalent of each row is also reported for transparency.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Brooks, M. E., Dalal, D. K., & Nolan, K. P. (2014). Are common language effect sizes easier to understand than traditional effect sizes? *Journal of Applied Psychology, 99*(2), 332–340. doi:10.1037/a0034745

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kraemer, H. C., & Kupfer, D. J. (2006). Size of treatment effects and their importance to clinical research and practice. *Biological Psychiatry, 59*(11), 990–996. doi:10.1016/j.biopsych.2005.09.014

Liu, X. S., Carlson, R., & Kelley, K. (2019). Common language effect size for correlations. *The Journal of General Psychology, 146*(3), 325–338. doi:10.1080/00221309.2019.1585321

McGraw, K. O., & Wong, S. P. (1992). A common language effect size statistic. *Psychological Bulletin, 111*(2), 361–365. doi:10.1037/00332909.111.2.361

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

Vargha, A., & Delaney, H. D. (2000). A critique and improvement of the CL common language effect size statistics of McGraw and Wong. *Journal of Educational and Behavioral Statistics, 25*(2), 101–132. doi:10.3102/10769986025002101

See Also

smd, ci_smd, nnt_from_smd

Other effect size estimates: cliff_delta(), correction_for_attenuation(), eta_squared(), eta_squared_generalized(), eta_squared_partial(), expected_partial_r(), expected_r(), expected_smd(), nnt_from_smd(), omega_squared(), omega_squared_partial(), probability_of_superiority_pai proportion_of_superiority(), responder_analysis(), smd_trimmed()

Examples

```
# 1. Point estimate only.
cles(smd = 0.5)

# 2. With a noncentral t CI from sample sizes (preferred):
cles(smd = 0.5, n_1 = 50, n_2 = 50, conf_level = 0.95)

# 3. With a pre-computed CI on d:
cles(smd = 0.5, smd_lower = 0.20, smd_upper = 0.80)

# 4. CL at three reference d values:
cles(smd = 0.2)
cles(smd = 0.5)
cles(smd = 0.8)
```

---

cliff_delta *Cliff's δ Ordinal Effect Size*

Description

Computes Cliff's (1993) δ statistic for two independent groups, the difference between the probability that a randomly drawn observation from group 1 exceeds one from group 2 and the reverse probability, together with an analytic confidence interval built from the U-statistic variance (Cliff, 1996). Most R implementations of Cliff's δ fall back to a bootstrap CI; the analytic CI here is faster, deterministic, and exact in the large-sample limit.

Usage

```
cliff_delta(group_1, group_2, conf_level = 0.95)
```

Arguments

group_1, group_2	Numeric vectors of observations in the two groups. Ordinal data are fine; the statistic uses only ranks.
conf_level	Confidence level for the CI. Default 0.95.

Details

Definition. Cliff's δ is

$$\delta \;=\; \Pr(Y_1 > Y_2) - \Pr(Y_1 < Y_2) \;=\; 2 \cdot A - 1,$$

where A is the Vargha-Delaney (2000) statistic. The sample estimator is $\hat{\delta} = (\#\{(i,j) : y_{1i} > y_{2j}\} - \#\{(i,j) : y_{1i} < y_{2j}\})/(n_1 n_2)$. Ties contribute zero to both counts. δ ranges over $[-1, 1]$, with 0 indicating no stochastic dominance.

Analytic CI. The asymptotic variance of $\hat{\delta}$ is (Cliff, 1993; restated as Feng & Cliff, 2004, Equation 2, p. 323)

$$\mathrm{Var}(\hat{\delta}) \;=\; \frac{(n_2 - 1)\sigma_{d_1}^2 + (n_1 - 1)\sigma_{d_2}^2 + \sigma_d^2}{n_1 n_2},$$

where $\sigma_{d_i}^2$ is the variance of the per-observation dominance scores within each group. (Feng & Cliff's printed equation transposes the $(n_1 - 1)$ and $(n_2 - 1)$ coefficients, which matters only for unequal group sizes; the pairing above is the correct one, checked by simulation against the empirical variance of $\hat{\delta}$.) The CI is constructed on the Fisher-style $\mathrm{arctanh}$-transformed scale and back-transformed to respect the bounded range of δ (analogous to Fisher's Z CI for Pearson r). Feng & Cliff (2004, Equation 5, p. 324) recommend an alternative asymmetric interval that models the dependence of the variance on δ; the two constructions agree to first order.

Connection to other measures. Cliff's δ is a linear transformation of the Vargha-Delaney (2000) A statistic ($\delta = 2A - 1$) and of the Mann-Whitney U statistic ($U/(n_1 n_2) = A$). It is the ordinal analog of the common-language effect size `cles` and is preferable when bivariate normality is implausible (skewed outcomes, ordinal scales).

Value

A `data.frame` with rows for the point estimate `cliff_delta` and the lower/upper CI bounds. The output also reports the proportion of pairs with $y_1 > y_2$, the proportion with $y_1 < y_2$, and the proportion of ties.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cliff, N. (1993). Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin, 114*(3), 494–509. doi:10.1037/00332909.114.3.494

Cliff, N. (1996). *Ordinal methods for behavioral data analysis*. Lawrence Erlbaum.

Feng, D., & Cliff, N. (2004). Monte Carlo evaluation of ordinal d with improved confidence interval. *Journal of Modern Applied Statistical Methods, 3*(2), 322–332. doi:10.22237/jmasm/1099267560

Long, J. D., Feng, D., & Cliff, N. (2003). Ordinal analysis of behavioral data. In I. B. Weiner (Ed.), *Handbook of psychology, Vol. 2: Research methods* (pp.\ 635–661). Wiley.

Vargha, A., & Delaney, H. D. (2000). A critique and improvement of the CL common language effect size statistics of McGraw and Wong. *Journal of Educational and Behavioral Statistics, 25*(2), 101–132. doi:10.3102/10769986025002101

See Also

cles, nnt_from_smd, ss_aipe_cliff_delta

Other effect size estimates: cles(), correction_for_attenuation(), eta_squared(), eta_squared_generalized(), eta_squared_partial(), expected_partial_r(), expected_r(), expected_smd(), nnt_from_smd(), omega_squared(), omega_squared_partial(), probability_of_superiority_paired(), proportion_of_superiorit responder_analysis(), smd_trimmed()

Examples

```
# 1. Two groups of different sizes, no ties:
set.seed(113)
a <- rnorm(30, mean = 0, sd = 1)
b <- rnorm(40, mean = 0.5, sd = 1)
cliff_delta(a, b)

# 2. With ties (ordinal data):
o1 <- c(1, 2, 2, 3, 3, 3, 4, 4, 5)
o2 <- c(2, 3, 3, 4, 4, 5, 5, 5)
cliff_delta(o1, o2)

# 3. Robust to right skew. Cliff's delta on the raw, untransformed
#    drinking outcome from the Smith, Meyers, and Delaney (1998)
#    trial, comparing the Community Reinforcement Approach (CRA)
#    against standard care. Because the statistic uses only ranks it
#    needs no normalizing transformation of the heavily skewed
#    outcome, unlike the standardized mean difference.
data(drinks_trial)
cra <- drinks_trial$drinks_per_week[drinks_trial$treatment == "CRA"]
std <- drinks_trial$drinks_per_week[drinks_trial$treatment == "Standard"]
cliff_delta(cra, std)
```

---

cohen_f *Cohen's* f *Effect Size*

Description

Computes Cohen's $f = \sigma_m/\sigma$, the population standard deviation of means relative to the within-group standard deviation, by any of three equivalent specifications:

1. raw population means and within-group variance,
2. the population proportion of variance accounted for, η^2,
3. σ_m and σ directly.

Cohen's f is a population quantity; supplied with population parameters it returns the population value, supplied with sample estimates it returns the corresponding sample value.

Usage

```
cohen_f(
  mu = NULL,
  sigma_squared = NULL,
  n = NULL,
  eta_squared = NULL,
  sigma_m = NULL,
  sigma = NULL
)
```

Arguments

mu	Numeric vector of population means (one per group). Use together with sigma_squared.
sigma_squared	The within-group variance. Use together with mu.
n	Optional. Per-group sample sizes (a single number for equal group sizes, or a vector of length length(mu) for unequal). When NULL, equal weighting across groups is used (i.e., the population variance of mu is computed with the $1/k$ divisor).
eta_squared	The population proportion of variance accounted for. Use this argument alone.
sigma_m	The population standard deviation of the means (σ_m). Use together with sigma.
sigma	The within-group standard deviation (σ). Use together with sigma_m.

Details

All three calling modes return the same value when applied to compatible inputs (Cohen 1988, eq. 8.2.1):

- Raw form: $f = \sqrt{\sum n_j(\mu_j - \bar{\mu})^2/N \cdot 1/\sigma^2}$.
- From η^2: $f = \sqrt{\eta^2/(1-\eta^2)}$.
- From the variance ratio: $f = \sigma_m/\sigma$.

Value

A 1-row data.frame with columns term and value; term is "cohen_f" and value is the computed value.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

See Also

ci_srsnr, ci_snr, ss_power_R2

Examples

```
# (1) From raw means and within-group variance:
cohen_f(mu = c(94, 91, 92, 83), sigma_squared = 67.375)

# Equal n weights are the default; equivalent with explicit equal n:
cohen_f(mu = c(94, 91, 92, 83), sigma_squared = 67.375, n = 6)

# Unequal n:
cohen_f(mu = c(94, 91, 92, 83), sigma_squared = 67.375, n = c(4, 6, 5, 5))

# (2) From eta_squared:
cohen_f(eta_squared = 0.10)

# (3) From sigma_m and sigma directly:
cohen_f(sigma_m = 4, sigma = 8)
```

---

cohen_h *Cohen's* h *Effect Size for a Difference Between Two Proportions*

Description

Computes Cohen's *h*, the effect size for the difference between two proportions on the arcsine (variance-stabilizing) scale,

$$h = \varphi_1 - \varphi_2, \qquad \varphi_i = 2\,\arcsin\sqrt{p_i}.$$

The arcsine transform spaces proportions so that a given *h* carries the same detectability wherever the proportions sit, which a raw difference $p_1 - p_2$ does not: a shift from .01 to .05 is easier to detect than one from .41 to .45, and *h* reflects that while the raw difference does not. Cohen's *h* is the proportion analogue of the standardized mean difference (`smd`): the effect size on which power and sample size planning for a difference between two proportions is conventionally based.

Usage

```
cohen_h(p1, p2)
```

Arguments

p1, p2	The two proportions, each in $[0, 1]$. h is $\varphi(p_1) - \varphi(p_2)$, so it is positive when p1 is the larger.

Details

Cohen's h is a population quantity: supplied with population proportions it returns the population value, supplied with sample proportions it returns the corresponding sample value. It is signed, positive when p1 exceeds p2; its magnitude `abs()` is the size of the effect irrespective of direction.

Value

A 1-row `data.frame` with columns `term` and `value`; `term` is `"cohen_h"` and `value` is the signed effect size.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum. (See Chapter 6.)

See Also

`smd` for the standardized mean difference, `cohen_f`, `ci_proportion`

Examples

```
# A shift from .40 to .55.
cohen_h(p1 = 0.55, p2 = 0.40)

# The same raw difference near the floor is a larger h, since a difference is
# easier to detect where the proportions are small.
cohen_h(p1 = 0.20, p2 = 0.05)

# Its magnitude is the size irrespective of direction.
abs(cohen_h(p1 = 0.40, p2 = 0.55)$value)
```

---

cohen_kappa *Cohen's Kappa Coefficient of Inter-Rater Agreement*

Description

Computes Cohen's (1960) kappa coefficient of agreement between two raters on a categorical variable, with optional weighting (linear, quadratic, or custom) for ordinal categories, following Cohen's (1968) weighted kappa. Input can be the two raters' classification vectors or a published $k \times k$ frequency table. A confidence interval and a Wald test of $H_0 : \kappa = 0$ are returned, using the asymptotic standard error of Fleiss, Cohen, and Everitt (1969).

Usage

```
cohen_kappa(
  rater_1 = NULL,
  rater_2 = NULL,
  table = NULL,
  weights = "unweighted",
```

```
  weight_scaling = c("agreement", "disagreement"),
  categories = NULL,
  conf_level = 0.95,
  ci_method = c("wald", "percentile", "bca"),
  B = 10000L,
  seed = NULL
)
```

Arguments

`rater_1`	First rater's classification vector (categorical, factor or coercible to factor).
`rater_2`	Second rater's classification vector (same length as `rater_1`; categorical, factor or coercible to factor). Together `rater_1` and `rater_2` give the two raters' classifications on the same set of subjects; pairs in which either is `NA` are dropped before computation.
`table`	A $k \times k$ frequency table of joint assignments (rows are rater 1's categories, columns rater 2's, in the same order), supplied instead of `rater_1` and `rater_2`. This is the form in which published agreement studies usually report their data. Row and column `dimnames`, when present, must match and are used as the category labels.
`weights`	Either the string `"unweighted"` (the default; appropriate for nominal categories), `"linear"`, or `"quadratic"`, or a user-supplied square numeric weight matrix. A custom matrix is interpreted on the scale named by `weight_scaling`. Asymmetric matrices are allowed; see *Details*.
`weight_scaling`	How a custom `weights` matrix is scaled: `"agreement"` (the default; diagonal 1, decreasing off-diagonal entries, typically in $[0, 1]$) or `"disagreement"` (Cohen's 1968 ratio-scaled disagreement weights: zero diagonal, larger entries for graver disagreements). The two scalings give identical κ_W; see *Details*.
`categories`	Optional character vector listing the full category set in display order (used both to set the order of the confusion matrix's rows/columns and to map ordinal categories to integers $1, \ldots, k$ for linear and quadratic weighting). When `NULL` (the default) and both raters are factors with the same level set, the factor levels are used in their own order; otherwise the sorted union of values observed in `rater_1` and `rater_2` is used. When supplied, the set must be unique, non-missing, and contain every observed rating; an omitted category raises an error, because silently dropping the corresponding ratings while still dividing by the original sample size would corrupt kappa. With `table` input, supply `categories` only when the table has no `dimnames`.
`conf_level`	Confidence level for the interval (default `0.95`).
`ci_method`	Interval method: `"wald"` (the default, the asymptotic interval from the Fleiss, Cohen, and Everitt standard error), `"percentile"` (bootstrap percentile), or `"bca"` (bootstrap bias-corrected and accelerated).
`B`	Number of bootstrap replications when `ci_method` is `"percentile"` or `"bca"` (default `10000`; ignored for `"wald"`). The BCa adjustment pushes the working quantiles into the tails, so reduce `B` for exploration, not for a reported analysis.

`seed` Optional integer seed for the bootstrap. The default `NULL` uses the current state of the random number generator; a supplied seed is set internally and the prior state restored on exit.

Details

For two raters and a confusion matrix P of joint proportions (rater 1 $\times$ rater 2), the weighted kappa is

$$\kappa_W = \frac{p_o^{(W)} - p_e^{(W)}}{1 - p_e^{(W)}}, \quad p_o^{(W)} = \sum_{i,j} W_{ij}\, P_{ij}, \quad p_e^{(W)} = \sum_{i,j} W_{ij}\, p_{i\cdot}\, p_{\cdot j},$$

where $p_{i\cdot}$ and $p_{\cdot j}$ are the row and column marginals. For `weights = "unweighted"` (the diagonal of W is 1, off- diagonal 0) this collapses to Cohen's original formulation.

Disagreement scaling. Cohen (1968) develops weighted kappa by ratio scaling *disagreement*: each cell receives a weight $v_{ij} \geq 0$, zero on the agreement diagonal, with, for example, a weight of 6 representing twice as much disagreement as 3. The weights are part of the definition of agreement (and of any hypothesis tested about it), so they must be fixed before the data are collected. κ_W is invariant to multiplying the v_{ij} by any positive constant, and a disagreement matrix is related to an agreement matrix by $w_{ij} = 1 - v_{ij}/v_{\max}$ (Cohen, 1968, Footnote 3), which is the conversion applied internally when `weight_scaling = "disagreement"`. Either scaling therefore yields the same κ_W; supply whichever is more natural.

Asymmetric weights and validity. Nothing in κ_W requires $W_{ij} = W_{ji}$. Symmetric weights suit reliability, where the two sources have equal status; asymmetric weights suit validity, where one source is a criterion and the other a predictor and the two directions of a confusion can carry different costs (Cohen, 1968). The examples reproduce Cohen's computer-diagnosis illustration.

Standard error. The Fleiss-Cohen-Everitt (1969) asymptotic variance for weighted kappa is used:

$$\mathrm{Var}(\hat\kappa_W) = \frac{1}{N(1-p_e^{(W)})^2}\Big[\sum_{i,j} P_{ij}\big(W_{ij} - (\bar W_{i\cdot} + \bar W_{\cdot j})(1-\hat\kappa_W)\big)^2 - \big(\hat\kappa_W - p_e^{(W)}(1-\hat\kappa_W)\big)^2\Big],$$

with $\bar W_{i\cdot} = \sum_j W_{ij}\, p_{\cdot j}$ and $\bar W_{\cdot j} = \sum_i W_{ij}\, p_{i\cdot}$. The Wald confidence interval is $\hat\kappa \pm z_{1-\alpha/2}\,\widehat{\mathrm{SE}}$. Cohen's (1968) own Formulas 10 and 13 for the standard error of κ_W preceded this result and were superseded by it; the examples reproduce his Table 1 arithmetic for the historical record while the function reports the Fleiss-Cohen-Everitt interval.

Choice of weights. Use `"unweighted"` for nominal categories. For ordinal categories, `"quadratic"` is the most common choice (and mathematically equivalent to the intraclass correlation under certain conditions; Fleiss & Cohen, 1973); `"linear"` is also defensible. Cohen (1968) further shows that with equal marginals and quadratic-pattern disagreement weights, κ_W equals the product-moment correlation between the category scores.

Small samples and the bootstrap. The Wald interval can have poor coverage for small N or extreme values of $\hat\kappa$; a bootstrap interval is more dependable in those regimes (Blackman & Koval, 2000). With `ci_method = "percentile"` or `"bca"` the subjects (the rated pairs) are resampled with replacement `B` times, kappa is recomputed on each resample with the same categories and weights, and the interval is read off the bootstrap distribution: the percentile interval takes the empirical quantiles, and the BCa interval adjusts the quantile positions for median bias (estimated from the bootstrap distribution) and for acceleration (estimated from the jackknife), making it second-order accurate where the percentile interval is first-order accurate (Efron & Tibshirani, 1993). `table`

input is expanded to the equivalent paired ratings and resampled the same way. A resample on which kappa is undefined (chance agreement 1) is dropped, and the interval is computed from the replications that return a value; a single warning reports how many were dropped. The `se`, `z_value`, and `p_value` columns keep their asymptotic definitions under every `ci_method`; only the interval changes. Bootstrap results vary from run to run; supply `seed` for reproducibility (the RNG state is set locally and the caller's state restored on exit).

Value

A one-row `data.frame` (class `dmar_tbl`) with columns `weights` (the form used), `kappa`, `se` (asymptotic standard error), `lower_limit`, `upper_limit`, `z_value`, `p_value` (Wald test of $H_0 : \kappa = 0$), `n` (number of paired ratings), and `n_categories` (k).

The per-cell detail behind the coefficient travels with the result as the `cells` attribute, in the form of Cohen's (1968) Table 1: a `data.frame` with one row per cell of the confusion matrix giving `rater_1` and `rater_2` (the cell's categories), `observed_proportion`, `expected_proportion` (the product of the marginal proportions, the cell's chance expectation), `weight` (the agreement-scale weight used in the computation), and, when `weight_scaling = "disagreement"`, the supplied `disagreement_weight`. Retrieve it with `attr(result, "cells")`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Blackman, N. J.-M., & Koval, J. J. (2000). Interval estimation for Cohen's kappa as a measure of agreement. *Statistics in Medicine, 19*(5), 723–741.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20*(1), 37–46.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin, 70*(4), 213–220.

Fleiss, J. L., Cohen, J., & Everitt, B. S. (1969). Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin, 72*(5), 323–327.

Fleiss, J. L., & Cohen, J. (1973). The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement, 33*(3), 613–619.

See Also

`diagnosis_agreement` (Cohen's 1968 Table 1 as a data set), `fleiss_kappa`, `icc`

Other reliability: `diagnosis_agreement`, `fleiss_kappa()`, `icc()`, `reliability()`, `reliability_H()`, `reliability_alpha()`, `reliability_kr20()`, `reliability_omega()`, `reliability_omega_categorical()`

Examples

```
# ------------------------------------------------------------------------
# Cohen (1968, Table 1): two judges assign N = 200 cases to three
# diagnostic categories. The table ships as the diagnosis_agreement
```

```
# data set in the paper's own layout, Judge B in rows and Judge A in
# columns, carrying Cohen's per-cell disagreement weights, observed
# proportions, and chance-expected proportions.
data(diagnosis_agreement)
tab <- xtabs(frequency ~ judge_b + judge_a, data = diagnosis_agreement)
v   <- unclass(xtabs(disagreement_weight ~ judge_b + judge_a,
                     data = diagnosis_agreement))
tab

# Unweighted kappa, all disagreements equal (Cohen's Formula 4): .492.
cohen_kappa(table = tab)

# A percentile bootstrap interval for the same table, which expands
# it to its 200 paired ratings and resamples the subjects. The point
# estimate and the asymptotic standard error are unchanged; only the
# interval is read off the bootstrap distribution. B = 2000 keeps the
# example quick; a reported interval deserves the default B = 10000.
cohen_kappa(table = tab, ci_method = "percentile", B = 2000,
            seed = 113)

# Cohen's ratio-scaled disagreement weights: a neurosis-psychosis
# confusion (weight 6) is six times as grave as a personality
# disorder-neurosis confusion (weight 1). Weighted kappa = .348,
# smaller than the unweighted .492: these judges disagree less than
# chance expectation where it matters little and at about the chance
# level where it matters most.
res <- cohen_kappa(table = tab, weights = v,
                   weight_scaling = "disagreement")
res

# The per-cell quantities of Cohen's Table 1 travel with the result:
# observed and chance-expected proportions and both weight scalings.
attr(res, "cells")

# Interchanging the 6 and 1 weights reverses the story: kappa_w = .574.
v_swap <- v
v_swap[v == 6] <- 1
v_swap[v == 1] <- 6
cohen_kappa(table = tab, weights = v_swap,
            weight_scaling = "disagreement")

# Cohen's own Table 1 arithmetic (his Formulas 8, 10, and 13),
# computed straight from the data set's per-cell columns and
# reproduced for the historical record. The function reports the
# Fleiss-Cohen-Everitt (1969) standard error, which superseded
# Formulas 10 and 13.
q_o <- with(diagnosis_agreement,
            sum(disagreement_weight * observed_proportion))   # = .90
q_c <- with(diagnosis_agreement,
            sum(disagreement_weight * expected_proportion))   # = 1.38
1 - q_o / q_c                          # kappa_w = .348   (Formula 8)
v2_o <- with(diagnosis_agreement,
             sum(disagreement_weight^2 * observed_proportion))
```

```
v2_c <- with(diagnosis_agreement,
             sum(disagreement_weight^2 * expected_proportion))
sqrt((v2_o - q_o^2) / (200 * q_c^2))   # = .0901   (Formula 10)
sqrt((v2_c - q_c^2) / (200 * q_c^2))   # = .0916   (Formula 13)
(1 - q_o / q_c) + c(-1, 1) * qnorm(0.975) * 0.0901  # 95% CI [.171, .524]
(1 - q_o / q_c) / 0.0916               # z = 3.80, p < .001

# Cohen's validity reinterpretation: Judge A is a diagnostic panel
# (the criterion), Judge B a computer diagnosis (the predictor), and
# the weights are asymmetric because the two directions of a
# confusion carry different costs. Oriented to the table's layout
# (rows = Judge B = computer, columns = Judge A = panel), the weights
# below reproduce Cohen's published quantities: sum(v * p_o) = .86,
# sum(v * p_c) = 1.33, kappa_w = .353, with his Formulas 10 and 13
# giving .0887 and .0915. The weighted kappa vignette works this
# example in full, including the orientation of the paper's printed
# weight display relative to these values.
v_validity <- matrix(
  c(0, 1, 2,
    1, 0, 2,
    4, 6, 0), nrow = 3, byrow = TRUE)
cohen_kappa(table = tab, weights = v_validity,
            weight_scaling = "disagreement")

# ------------------------------------------------------------------------
# Raw rater vectors and ordinal categories (quadratic agreement
# weights, e.g., Likert-style severity).
set.seed(113)
x <- sample(1:5, 100, replace = TRUE)
y <- pmin(pmax(x + sample(-1:1, 100, replace = TRUE), 1), 5)
cohen_kappa(x, y, weights = "quadratic")
```

`combine_p` *Combine Independent P-Values Across Studies*

Description

Combines the one-tailed p-values from k independent tests of a common directional hypothesis into a single test, by any of the four classical methods that Raudenbush (1984) applied to the teacher expectancy experiments: Fisher's (1938) "adding logs" chi square, Edgington's (1972) "adding ps", the Mosteller and Bush (1954) "adding Zs" (Stouffer) method, and a weighted adding-Zs variant (weights are typically the studies' degrees of freedom). Combined significance tests answer the narrow question "is there an effect in at least some studies?"; they do not estimate its size. Pair them with `meta_smd` or `meta_es` for estimation, which is almost always the more informative summary.

Usage

```
combine_p(
  p,
```

```
  method = c("fisher", "edgington", "stouffer", "stouffer_weighted"),
  weights = NULL
)
```

Arguments

p	Numeric vector of one-tailed *p*-values, each in (0, 1), oriented so that small values support the common directional hypothesis.
method	Character vector naming the methods to compute: any of "fisher", "edgington", "stouffer", "stouffer_weighted"; the default computes all four (the "stouffer_weighted" row appears only when weights is supplied).
weights	Optional non-negative weights for "stouffer_weighted", one per study; degrees of freedom are the conventional choice (Mosteller & Bush, 1954).

Details

Fisher's statistic is $-2\sum \log p_i$, distributed chi square with $2k$ degrees of freedom under the joint null. Edgington's statistic is the plain sum $\sum p_i$, referred to a normal approximation with mean $k/2$ and variance $k/12$ (accurate for $k \geq 10$; for smaller k it is conservative in the tails). The Stouffer statistic is $\sum z_i/\sqrt{k}$ with $z_i = \Phi^{-1}(1-p_i)$, and the weighted variant is $\sum w_i z_i/\sqrt{\sum w_i^2}$. All four are reported with one-tailed combined *p*-values, matching the directional inputs.

Methods can disagree, and the disagreement is informative: Rosenthal (1978) notes there is no uniformly best test. In the published analysis, Raudenbush (1984) found three of the four rejecting the null at the .05 level while the df-weighted variant did not, an early warning that large studies were finding smaller effects. Computed from the study-level *p*-values as tabled, the example below shows two of the four rejecting: Fisher's ($p = .004$) and Stouffer's ($p = .014$) tests reject, Edgington's sits just above the level ($p = .051$; the tabled values sum to 7.00 where the paper's Table 2, p. 90, prints a sum of 6.84 with $p = .04$), and the df-weighted variant is not close ($p = .192$).

Value

A data.frame (class dmar_tbl) with, per requested method, its statistic row(s) and a one-tailed <method>_p row, plus a final k row. The *p* rows print to fixed decimals via the p_terms attribute.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Edgington, E. S. (1972). An additive method for combining probability values from independent experiments. *The Journal of Psychology, 80*(2), 351–363.

Fisher, R. A. (1938). *Statistical methods for research workers* (7th ed.). Oliver & Boyd.

Mosteller, F., & Bush, R. R. (1954). Selected quantitative techniques. In G. Lindzey (Ed.), *Handbook of social psychology* (Vol. 1). Addison-Wesley.

Raudenbush, S. W. (1984). Magnitude of teacher expectancy effects on pupil IQ as a function of the credibility of expectancy induction: A synthesis of findings from 18 experiments. *Journal of Educational Psychology, 76*(1), 85–97.

Rosenthal, R. (1978). Combining results of independent studies. *Psychological Bulletin, 85*(1), 185–193.

See Also

meta_smd and meta_es for estimating the pooled effect rather than only testing it; meta_contrast for differences among study effects; teacher_expectancy for the data behind the examples.

Other meta-analysis: meta_contrast(), meta_es(), meta_r(), meta_smd(), plot_forest()

Examples

```
# Raudenbush (1984), Table 2: the four combined tests over the 18
# teacher expectancy studies (Pellegrini & Hicks at its study-level
# values), weighting the Z method by degrees of freedom.
data(teacher_expectancy)
study <- teacher_expectancy[-c(4, 5), ]
p18  <- append(study$p_one_tailed, .010, after = 3)
df18 <- append(study$n_experimental + study$n_control - 2, 42, after = 3)
combine_p(p18, weights = df18)
# Fisher chi square 62.17 on 36 df; Edgington sum near 7; Stouffer
# z near 2.2; and the df-weighted z under 1: the large studies disagree.
```

---

common_method_marker *Marker-Variable Adjustment for Common Method Variance*

Description

The marker-variable technique of Lindell and Whitney (2001) estimates common method variance from the correlation of a *marker* variable that is theoretically unrelated to at least one of the substantive variables: any non-zero correlation it shows with that variable is attributed to shared method, and that amount is partialled out of the substantive correlations. When no a priori marker is available, the smallest positive correlation among the substantive items is used as a proxy, the common marker-free variant of the method. A correlation that remains statistically significant after the adjustment, by the paper's t test of the adjusted correlation with $N - 3$ degrees of freedom (their Equation 5), is evidence that the relationship is not an artifact of method variance; the test is applied by the user, since this function works from the correlation matrix alone and does not take N.

Usage

```
common_method_marker(R, marker_r = NULL)
```

Arguments

R	A correlation matrix among the substantive items.
marker_r	The marker variable's (CMV) correlation. When NULL (default) the smallest positive off-diagonal correlation in R is used as the proxy marker.

Details

Writing r_M for the marker (or proxy) correlation, each substantive correlation is adjusted as $r_{ij}^A = (r_{ij} - r_M)/(1 - r_M)$ (Lindell & Whitney, 2001, Equation 4). The CMV-adjusted correlation matrix is returned as the `"adjusted"` attribute; the reported table summarizes the marker correlation and the average absolute correlation before and after adjustment.

The method presumes the variables are reflected so that their intercorrelations are positive; a negative substantive correlation is pushed further from zero by the adjustment rather than attenuated, so reverse-code as needed before adjusting.

Value

A `data.frame` (class `dmar_tbl`) with rows `marker_correlation`, `mean_abs_r_unadjusted`, and `mean_abs_r_adjusted` in the `value` column. The full adjusted correlation matrix is the `"adjusted"` attribute.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Lindell, M. K., & Whitney, D. J. (2001). Accounting for common method variance in cross-sectional research designs. *Journal of Applied Psychology, 86*(1), 114–121. doi:10.1037/0021-9010.86.1.114

See Also

`common_method_single_factor` for the single-factor screen.

Other multivariate and latent variable methods: `average_variance_extracted()`, `bifactor_indices()`, `cfa_1()`, `cfa_2()`, `cfa_k()`, `ci_eigenvalue()`, `common_method_single_factor()`, `dmacs()`, `ecvi()`, `htmt()`, `irt_grm()`, `irt_information()`, `measurement_alignment()`, `measurement_invariance()`, `procrustes_phi()`, `simple_structure()`

Examples

```
R <- matrix(c(1, .5, .4, .5, 1, .45, .4, .45, 1), 3, 3,
            dimnames = list(c("a", "b", "c"), c("a", "b", "c")))
res <- common_method_marker(R, marker_r = 0.10)
res
attr(res, "adjusted")
```

common_method_single_factor *Single-Common-Factor Screen for Common Method Variance*

Description

This function implements Harman's single-factor test, the most widely used (and weakest) screen for common method variance: fit a one-factor model to all of the items by maximum likelihood and inspect how much of their variance the common factor accounts for. The rationale is that if a single method factor dominated the responses, one common factor would capture a large share of the variance. A factor accounting for more than half of the variance is the customary red flag (Podsakoff, MacKenzie, Lee, & Podsakoff, 2003). The screen is coarse and cannot by itself rule method variance in or out; the marker-variable and latent method factor approaches are stronger (see common_method_marker).

Usage

```
common_method_single_factor(data = NULL, S = NULL, R = NULL)
```

Arguments

data	A data.frame or numeric matrix of item responses. Supply this, a covariance matrix S, or a correlation matrix R (exactly one).
S	A symmetric covariance matrix among the items, when raw data are not available but the summary statistics a paper reports are. It is converted to a correlation matrix internally, so the test acts on the same scale-free quantity regardless of which input is supplied.
R	A correlation matrix among the items, when raw data are not available.

Details

Harman's single-factor test (the proportion of variance explained by one common factor) is related to but distinct from the marker-variable technique. A *marker variable* (or common-method marker) is a variable chosen to be theoretically unrelated to the substantive constructs under study, so that any observed correlation between it and the substantive items is attributable to shared method rather than to a true relationship; it is used to estimate or partial out common method variance (Lindell & Whitney, 2001). The single-factor test uses no such marker, it asks only whether a single dimension dominates the item set, so it can flag a strong common factor but cannot identify whether that factor is method or substance.

The one-factor model is fit to the item correlation matrix by maximum likelihood with factanal, and the statistic is the proportion of total variance the common factor accounts for: the sum of the squared standardized loadings divided by the number of items (equivalently, the mean communality). Much of the applied literature computes the screen from the largest eigenvalue of the correlation matrix, which describes the first principal component, not a factor; the test is implemented factor analytically here, in the psychometric tradition, because a principal component absorbs unique as well as common variance and so overstates the share a common factor accounts for.

Correlations from raw data use pairwise-complete observations. A supplied covariance matrix is first standardized to a correlation matrix with `cov2cor`. The one-factor model requires at least three items.

Value

A `data.frame` (class `dmar_tbl`) with rows `variance_explained` (the proportion of total variance the single common factor accounts for) and `n_items` in the `value` column.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Harman, H. H. (1976). *Modern factor analysis* (3rd ed.). University of Chicago Press.

Lindell, M. K., & Whitney, D. J. (2001). Accounting for common method variance in cross-sectional research designs. *Journal of Applied Psychology, 86*(1), 114–121. doi:10.1037/0021-9010.86.1.114

Podsakoff, P. M., MacKenzie, S. B., Lee, J.-Y., & Podsakoff, N. P. (2003). Common method biases in behavioral research: A critical review of the literature and recommended remedies. *Journal of Applied Psychology, 88*(5), 879–903. doi:10.1037/00219010.88.5.879

See Also

`common_method_marker` for the marker-variable adjustment.

Other multivariate and latent variable methods: `average_variance_extracted()`, `bifactor_indices()`, `cfa_1()`, `cfa_2()`, `cfa_k()`, `ci_eigenvalue()`, `common_method_marker()`, `dmacs()`, `ecvi()`, `htmt()`, `irt_grm()`, `irt_information()`, `measurement_alignment()`, `measurement_invariance()`, `procrustes_phi()`, `simple_structure()`

Examples

```
set.seed(113)
f <- rnorm(200)
d <- data.frame(
  x1 = f + rnorm(200), x2 = f + rnorm(200), x3 = f + rnorm(200),
  x4 = rnorm(200),     x5 = rnorm(200),     x6 = rnorm(200))
common_method_single_factor(d)

# The same screen from the summary statistics a paper reports.
common_method_single_factor(S = cov(d))
```

compare_cov_structures *Likelihood-Ratio Comparison of Covariance Structures*

Description

Fits a long-format within-subjects regression under a menu of variance-covariance structures, from independence through the unstructured form, and returns a comparison table of log-likelihood, AIC, BIC, and pairwise likelihood-ratio tests against the most general structure (UN). Wraps `gls`.

Usage

```
compare_cov_structures(
  data,
  outcome,
  subject,
  time,
  fixed_effects = NULL,
  structures = c("IND", "CS", "CSH", "AR1", "ARH1", "TOEP", "TOEPH", "UN")
)
```

Arguments

`data`	Long-format `data.frame` with one row per subject-by-condition observation.
`outcome`	Character name of the response column.
`subject`	Character name of the subject-id column.
`time`	Character name of the time / within-subjects factor column.
`fixed_effects`	Right-hand-side formula for the fixed effects (default: `~ time`).
`structures`	Character vector of structures to fit. Any subset of `c("IND", "CS", "CSH", "AR1", "ARH1", "TOEP", "TOEPH", "UN")` (default: all eight). Matching is case insensitive, so lowercase aliases such as `"cs"`, `"ar1"`, `"csh"`, `"arh1"`, `"toep"`, and `"un"` are accepted and normalized to their canonical uppercase labels.

Details

Structures. Every structure below is nested in UN, so the likelihood-ratio test against UN is well defined for each.

- `IND`: independent observations within subject (`correlation = NULL` in `gls`). Provided as a baseline.
- `CS`: compound symmetry, a constant correlation and a single variance across time points: `nlme::corCompSymm()`.
- `CSH`: heterogeneous compound symmetry, a constant correlation with a separate variance at each time point: `nlme::corCompSymm()` with `nlme::varIdent()`.
- `AR1`: first-order autoregressive correlation with a single variance: `nlme::corAR1()`.

- ARH1: heterogeneous first-order autoregressive correlation with a separate variance at each time point: nlme::corAR1() with nlme::varIdent().
- TOEP: Toeplitz (banded), a separate correlation at each lag with a single variance: nlme::corARMA() with autoregressive order one less than the number of time points and no moving-average term.
- TOEPH: heterogeneous Toeplitz, the Toeplitz correlation with a separate variance at each time point: nlme::corARMA() with nlme::varIdent().
- UN: unstructured, every variance and covariance free: nlme::corSymm() with nlme::varIdent().

LRT. Each restricted structure is compared against UN by the likelihood-ratio test. Both fits are re-estimated under ML (not REML) for the LRT, following nlme convention. The chi square statistic is $-2(\log L_{\text{restricted}} - \log L_{\text{UN}})$ on degrees of freedom equal to the difference in parameter count.

Caveats. The likelihood-ratio test against UN is valid because each listed structure is a restriction of UN. Two structures that are not nested in each other (for example CS and AR(1)) should be compared by AIC or BIC rather than by an LRT.

Value

A data.frame with one row per structure. Columns: structure, log_lik, AIC, BIC, n_par, LRT_vs_UN_chisq, LRT_vs_UN_df, LRT_vs_UN_p.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., & Schabenberger, O. (2006). *SAS for mixed models* (2nd ed.). SAS Institute.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 15.)

Pinheiro, J. C., & Bates, D. M. (2000). *Mixed-effects models in S and S-PLUS*. Springer.

See Also

gls, logLik, anova

Other hypothesis tests: adjusted_means(), ancova(), anova_within(), ci_dunnett(), ci_scheffe(), ci_tukey_kramer(), contrast_test(), correlations_test(), equivalence_r(), equivalence_smd(), factorial_anova(), manova_split_plot(), mauchly_test(), mixed_anova(), obrien_test(), pairwise_within(), randomization_test(), randomization_test_paired(), regions_of_significance(), simple_effects_AB(), summary_t_test(), welch_t()

Examples

```
# Four repeated measures on each of 30 subjects.
set.seed(113)
n <- 30; k <- 4
subj <- factor(rep(1:n, each = k))
tm   <- factor(rep(1:k, times = n))
```

```
y    <- as.vector(t(matrix(rnorm(n * k), n, k) +
                    rep(rnorm(n, 0, 1), each = k)))
d <- data.frame(y, subj, tm)

# All eight structures at once. Read the table by comparing AIC and BIC
# across rows, and use the likelihood-ratio test only for the nested
# comparison it reports, each structure against UN.
compare_cov_structures(d, outcome = "y", subject = "subj",
                       time = "tm")

# A subset, requested with lowercase aliases (matching is case
# insensitive).
compare_cov_structures(d, outcome = "y", subject = "subj",
                       time = "tm",
                       structures = c("cs", "csh", "ar1", "arh1"))
```

`content_validity_index` *Content Validity Index From Expert Ratings*

Description

Quantifies how well a pool of candidate items covers the construct it is meant to measure, using the relevance ratings of a panel of subject matter experts. Validation begins before any respondent data are collected: experts rate each item for relevance, and the content validity index summarizes their agreement (Lynn, 1986; Polit & Beck, 2006; Bandalos, 2018). The function returns the item level index (I-CVI) with an exact binomial confidence interval, the chance corrected modified kappa of Polit, Beck, and Owen (2007), Lawshe's (1975) content validity ratio, and the two scale level summaries S-CVI/Ave and S-CVI/UA. The confidence interval is what keeps a small panel from being read as more informative than it is: with five experts an I-CVI of 0.80 carries an interval roughly seven tenths of the width of the scale.

Usage

```
content_validity_index(
  ratings,
  relevant = c(3, 4),
  essential = NULL,
  conf_level = 0.95
)
```

Arguments

`ratings`	A matrix or `data.frame` of expert ratings with items in rows and experts in columns, on the conventional 4 point relevance scale (1 = not relevant, 2 = somewhat relevant, 3 = quite relevant, 4 = highly relevant). Every non-missing entry must be one of 1, 2, 3, or 4. Row names, when present, name the items; otherwise items are labeled `item_1`, `item_2`, and so on. At least 1 item and at least 2

experts are required. NA is allowed and means that expert did not rate that item; a missing entry lowers that item's expert count and thereby the denominator of its I-CVI, kappa, and CVR, leaving other items untouched. A row with no ratings at all is an error.

`relevant` The rating values counted as relevant. Default `c(3, 4)`, the standard dichotomization of the 4 point scale into relevant (3 or 4) versus not relevant (1 or 2). Must be a subset of `1:4`.

`essential` Optional. The rating values counted as "essential" for Lawshe's content validity ratio, when the panel answered the essential-versus-not question on a separate part of the scale. Must be a subset of `1:4`. When `NULL` (default), the content validity ratio is computed from the same dichotomization as relevance, that is from `relevant`.

`conf_level` Confidence level for the exact binomial interval on each I-CVI. Default `0.95`. Must be in (0, 1).

Details

Let N be the number of experts who rated an item and A the number of those who rated it relevant.

Item level index. The item level content validity index is the proportion of rating experts who called the item relevant,

$$\text{I-CVI} = A/N.$$

Lynn (1986) gives the conventional criteria: with five or fewer experts an item is expected to reach 1.00, and with six to ten experts at least 0.78.

Confidence interval. The I-CVI is a binomial proportion, so the interval reported here is the exact (Clopper-Pearson) interval from `binom.test` at `conf_level`. Expert panels are small by design and the interval states plainly how little a handful of ratings pins down the population proportion. This is the accuracy in parameter estimation view of content validity: if the interval is too wide to act on, the remedy is more experts.

Chance corrected agreement. Some of the observed agreement on relevance would occur if experts responded at random with probability 0.5, so Polit, Beck, and Owen (2007) correct the index in the manner of a kappa. The probability of chance agreement is the binomial point probability

$$p_c = \frac{N!}{A!\,(N-A)!}\,0.5^N,$$

and the modified kappa is

$$\kappa = \frac{\text{I-CVI} - p_c}{1 - p_c}.$$

The binomial coefficient is evaluated on the log scale with `lchoose` and then exponentiated, so a large panel does not overflow the way `factorial()` would.

Content validity ratio. Lawshe (1975) asked a panel whether each item measures behavior that is essential to the performance domain. With n_e experts calling the item essential,

$$\text{CVR} = \frac{n_e - N/2}{N/2},$$

which equals 1 when every expert says essential, 0 when exactly half do, and -1 when none do.

Scale level summaries. S-CVI/Ave is the mean of the I-CVIs over items, the averaging approach Polit and Beck (2006) recommend reporting. S-CVI/UA is the universal agreement proportion, the fraction of items whose I-CVI equals 1, a stricter and considerably more conservative summary.

Value

A `data.frame` (class `dmar_tbl`) with one row per item, in the order the items appear in `ratings`, and columns:

`item` The item name, from the row names of `ratings` when present.

`n_experts` Number of experts who rated the item (missing ratings excluded).

`n_relevant` Number of those experts whose rating was in `relevant`.

`i_cvi` The item level content validity index, `n_relevant / n_experts`.

`ci_lower`, `ci_upper` Limits of the exact binomial confidence interval for `i_cvi` at `conf_level`.

`kappa` The modified kappa of Polit, Beck, and Owen (2007), the I-CVI adjusted for chance agreement.

`cvr` Lawshe's content validity ratio, computed from `essential` when supplied and from `relevant` otherwise.

Attributes: `"s_cvi_ave"`, the mean of the `i_cvi` column; `"s_cvi_ua"`, the proportion of items with `i_cvi` equal to 1; `"relevant"`, the rating values counted as relevant; `"essential"`, the rating values counted as essential for the content validity ratio (equal to `"relevant"` when `essential` was `NULL`); and `"conf_level"`, the confidence level used.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bandalos, D. L. (2018). *Measurement theory and applications for the social sciences*. Guilford Press.

Lawshe, C. H. (1975). A quantitative approach to content validity. *Personnel Psychology, 28*(4), 563–575.

Lynn, M. R. (1986). Determination and quantification of content validity. *Nursing Research, 35*(6), 382–385.

Polit, D. F., & Beck, C. T. (2006). The content validity index: Are you sure you know what's being reported? Critique and recommendations. *Research in Nursing and Health, 29*(5), 489–497. doi:10.1002/nur.20147

Polit, D. F., Beck, C. T., & Owen, S. V. (2007). Is the CVI an acceptable indicator of content validity? Appraisal and recommendations. *Research in Nursing and Health, 30*(4), 459–467. doi:10.1002/nur.20199

See Also

`gwet_ac`, `fleiss_kappa` for agreement among raters on a common set of units.

Other agreement and measurement: `R2_mixed_effects()`, `gwet_ac()`, `icc_lmer()`, `krippendorff_alpha()`, `limits_of_agreement()`, `lin_ccc()`, `variance_components_mls()`

Examples

```
# Six experts rate five candidate items on the 4 point relevance scale.
ratings <- rbind(
  item_1 = c(4, 4, 3, 4, 4, 3),
  item_2 = c(4, 3, 4, 4, 3, 2),
  item_3 = c(2, 3, 1, 2, 3, 2),
  item_4 = c(4, 4, 4, 4, 4, 4),
  item_5 = c(3, 4, 4, 3, NA, 4))
colnames(ratings) <- paste0("expert_", 1:6)
cvi <- content_validity_index(ratings)
cvi

# The scale level summaries travel with the table as attributes.
attr(cvi, "s_cvi_ave")
attr(cvi, "s_cvi_ua")

# The worked example of Polit, Beck, and Owen (2007): 6 experts, 5 of whom
# rate the item relevant, gives I-CVI = 0.83, p_c = 0.094, and a modified
# kappa of 0.816 (the paper reports 0.81, carrying its rounded I-CVI).
content_validity_index(matrix(c(4, 4, 3, 4, 3, 1), nrow = 1))

# A wide interval is the point: with 5 experts an I-CVI of 0.80 is
# compatible with a population proportion anywhere from about 0.28 to 0.99.
content_validity_index(matrix(c(4, 4, 3, 4, 1), nrow = 1))

# The broom verbs on the earlier result: one row per item, and the
# scale-level summary.
generics::tidy(cvi)
generics::glance(cvi)
```

`contrast_adjusted` *Confidence Interval for a Contrast of Covariate-Adjusted Cell Means in a Factorial ANCOVA*

Description

Given a fitted `lm` or `aov` object for a factorial analysis of covariance (one or more crossed factors plus one or more covariates) and a numeric contrast vector over the cells of the factorial design, `contrast_adjusted()` forms the contrast of the covariate-adjusted cell means, $\hat{\psi} = \sum_j c_j \hat{\bar{Y}}_j$, where each $\hat{\bar{Y}}_j$ is the model's predicted mean for cell j evaluated at the mean of every covariate (the adjusted, or least-squares, cell mean). It returns the point estimate, a *t* confidence interval on the model's residual degrees of freedom, and the accompanying *t* statistic and two-sided *p*-value for $H_0 : \psi = 0$.

Usage

```
contrast_adjusted(model, contrast, conf_level = 0.95)
```

Arguments

model	A fitted lm or aov object for a factorial ANCOVA: one or more crossed factors and one or more numeric covariates on the right-hand side of the formula.
contrast	A numeric vector of contrast weights, one weight per cell of the factorial design (the crossing of the model's factors). Its length must equal the number of cells. The weights typically sum to zero.
conf_level	The confidence level for the interval (default 0.95).

Details

The adjusted cell means are the means the ANCOVA actually tests: the predicted outcome for each combination of factor levels, holding every covariate at its sample mean. Writing L for the linear map that sends the model coefficients to that contrast of adjusted means (built by evaluating the model's design matrix at each cell with the covariates set to their means and combining the rows with the contrast weights), the point estimate is $\hat{\psi} = L'\hat{\beta}$ and its standard error is the square root of the quadratic form $L'\,\mathrm{vcov}(\hat{\beta})\,L$. The interval is $\hat{\psi} \pm t_{1-\alpha/2,\,\nu}\,\mathrm{SE}$, with ν the residual degrees of freedom of the fitted model.

Because L is read off the fitted model's own design matrix, the function is agnostic to how the factors are parameterized: a cell-means parameterization (y ~ 0 + cell + x) and the crossed-factor parameterization (y ~ A * B + x) give the same contrast estimate and standard error, provided the contrast vector is ordered to match the cells of the reference grid (see Note).

The *t* interval returned here has exact per-comparison coverage for a single contrast chosen in advance. For a family of contrasts examined together, adjust the critical value for multiplicity (for example the Scheffe critical value for the full cell space, cv_scheffe, or a Bryant–Paulson simultaneous interval, ci_c_ancova_bp).

Value

A five-row dmar_tbl (a data.frame with columns term and value). The term values are "contrast" (the point estimate $\hat{\psi}$ of the contrast of adjusted cell means), "lower_limit" and "upper_limit" (the confidence limits), "t" (the *t* statistic), and "p" (the two-sided *p*-value). The stored value column is numeric at full precision.

Note

The contrast weights are matched to the cells of the reference grid, which is the crossing of the model's factors in the order the factors appear in the model formula, with the first factor varying fastest (the order expand.grid produces over the factor levels). For a single factor this is simply the order of its levels. When in doubt, fit the cell-means form y ~ 0 + cell + covariates with cell = interaction(A, B, ...) and order the weights to match levels(cell).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9 on designs with covariates.)

See Also

contrast_test for contrasts of unadjusted group means in a one-way design; ci_c_ancova for a single-covariate ANCOVA contrast from summary statistics.

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), plot_smd()

Examples

```
# A 2 x 2 factorial ANCOVA with one covariate.
set.seed(113)
d <- data.frame(
  A = factor(rep(c("a1", "a2"), each = 40)),
  B = factor(rep(rep(c("b1", "b2"), each = 20), 2)),
  x = rnorm(80)
)
d$y <- 5 + 2 * (d$A == "a2") + 1.5 * (d$B == "b2") +
  0.8 * d$x + rnorm(80)
fit <- lm(y ~ A * B + x, data = d)

# Cells in reference-grid order: (a1,b1), (a2,b1), (a1,b2), (a2,b2).
# Main effect of A, averaged over B: mean(a2 cells) - mean(a1 cells).
contrast_adjusted(fit, contrast = c(-0.5, 0.5, -0.5, 0.5))
```

---

contrast_test *Tests One or More Contrasts of Group Means in a One-Way Design*

Description

Given a fitted one-way aov or lm object and a set of contrast weights, computes for every contrast the estimate $\hat{\psi} = \sum_i c_i \bar{Y}_i$, its standard error, *t*-statistic, degrees of freedom, two-sided *p*-value, and confidence interval. Supports several common multiple-comparison adjustments and either equal-variance (pooled) or Welch-style unequal-variance inference.

Usage

```
contrast_test(
  object,
  contrasts = "pairwise",
  adjust = "none",
```

```
  conf_level = 0.95,
  var_equal = TRUE
)
```

Arguments

`object`	A fitted `aov` or `lm` object for a one-way design (a single grouping factor on the right-hand side of the formula).
`contrasts`	Specification of one or more contrasts. Any of: `"pairwise"` **(default)** All pairwise comparisons among the group means. **a named list of numeric vectors** Each vector is one contrast and its name is used as the row label. **a numeric matrix** Each row is one contrast; `rownames`, if present, are used as labels. **a numeric vector** Treated as a single contrast. Each contrast vector must have length equal to the number of groups, and the weights are typically chosen to sum to zero.
`adjust`	Multiple-comparison adjustment. One of `"none"` (default), `"bonferroni"`, `"scheffe"`, `"tukey"` (pairwise contrasts only), or any of the sequential methods supported by `p.adjust` (`"holm"`, `"hochberg"`, `"BH"`, `"BY"`).
`conf_level`	Confidence level for the interval (default `0.95`).
`var_equal`	Logical. If `TRUE` (default), uses the pooled error variance MS_{error} and the residual degrees of freedom from `object`. If `FALSE`, uses each group's own sample variance and a Welch-Satterthwaite approximate df per contrast.

Details

Test statistic. For a contrast with weights $c_1, \ldots, c_k$ (k = number of groups), the estimate is $\hat{\psi} = \sum_i c_i \bar{Y}_i$. Under equal variances, the standard error is $\sqrt{MS_{\text{error}} \sum_i c_i^2 / n_i}$ with $df = N - k$; under unequal variances, the standard error is $\sqrt{\sum_i c_i^2 s_i^2 / n_i}$ with the Welch-Satterthwaite df,

$$df_{\text{Welch}} = \frac{\left(\sum_i c_i^2 s_i^2 / n_i\right)^2}{\sum_i (c_i^2 s_i^2 / n_i)^2 / (n_i - 1)}.$$

The unadjusted p-value is two-sided based on the t reference distribution.

Adjustments. The `p_adjusted` and confidence interval critical value are computed as follows.

- `"none"`: no adjustment; the CI uses $t_{1-\alpha/2, df}$.
- `"bonferroni"`: $p_{\text{adj}} = \min(1, m\,p)$ for m contrasts, with CI based on $t_{1-\alpha/(2m), df}$.
- `"scheffe"`: appropriate for any contrast (or family of contrasts). p_{adj} comes from the upper tail of an F reference distribution applied to $t^2/(k-1)$, and the CI uses $\sqrt{(k-1)\, F_{1-\alpha,\, k-1, df}}$.
- `"tukey"`: requires every contrast to be pairwise. Uses the studentized range distribution (`ptukey`/`qtukey`) so that $p_{\text{adj}} = 1 - \text{ptukey}(|t|\sqrt{2}; k, df)$ and the CI uses $q_{1-\alpha,\, k, df}/\sqrt{2}$.
- `"holm"`, `"hochberg"`, `"BH"`, `"BY"`: `p.adjust` is applied to the unadjusted p-values; the CI uses the unadjusted t-critical value because these methods do not give simultaneous CIs in closed form.

Variance assumption with adjustments. The Tukey and Scheffé procedures assume equal variances; combining them with `var_equal = FALSE` is at the user's risk (the resulting Type I error rate is no longer guaranteed). For unequal variances, common alternatives are Games-Howell (Tukey-style) and Brown-Forsythe (Scheffé-style); these are not currently supported here.

Scope. Only one-way designs are supported in v1 (one outcome, one grouping factor). Multi-way designs throw an informative error.

Value

A `data.frame` with one row per contrast and columns `contrast`, `estimate`, `se`, `t`, `df`, `p_value`, `p_adjusted`, `ci_lower`, and `ci_upper`. The adjustment, confidence level, and variance assumption are stored as `attr(*, "adjust")`, `attr(*, "conf_level")`, and `attr(*, "var_equal")`. The table prints through the `dmar_tbl` display layer and works with `tidy` and `glance` (see `dmar_tidiers`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Hsu, J. C. (1996). *Multiple comparisons: Theory and methods*. Chapman & Hall.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Scheffe, H. (1953). A method for judging all contrasts in the analysis of variance. *Biometrika, 40*, 87–104.

Tukey, J. W. (1953). The problem of multiple comparisons. Unpublished manuscript, Princeton University.

See Also

`TukeyHSD`, `pairwise.t.test`, `p.adjust`, `cv_tukey_hsd`, and `dmar_tidiers` for the tidy methods

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `correlations_test()`, `equivalence_r()`, `equivalence_smd()`, `factorial_anova()`, `manova_split_plot()`, `mauchly_test()`, `mixed_anova()`, `obrien_test()`, `pairwise_within()`, `randomization_test()`, `randomization_test_paired()`, `regions_of_significance()`, `simple_effects_AB()`, `summary_t_test()`, `welch_t()`

Examples

```
# All pairwise comparisons among the three arms of the depression_bdi
# treatment study.
fit <- aov(bdi_post ~ condition, data = depression_bdi)
contrast_test(fit, contrasts = "pairwise")

# Custom contrasts with a Tukey-protected family-wise error rate.
contrast_test(fit, contrasts = "pairwise", adjust = "tukey")

# A user-defined contrast: the SSRI arm vs. the average of the placebo
# and wait list arms. With levels ordered ssri, placebo, wait_list, the
```

```
# weights c(1, -0.5, -0.5) estimate that difference.
contrast_test(
  fit,
  contrasts = list("ssri vs non-drug arms" = c(1, -0.5, -0.5)),
  adjust = "scheffe"
)

# Welch-style inference: the wait list variance is about twice the
# SSRI variance, so the pooled error term is worth questioning.
contrast_test(fit, contrasts = "pairwise", var_equal = FALSE)

# Pairwise treatment comparisons in the Smith, Meyers, and Delaney
# (1998) drinking trial, on the normalizing log scale. Each row is
# one pairwise contrast of the three treatment means.
fit_drinks <- aov(log_drinks ~ treatment, data = drinks_trial)
contrast_test(fit_drinks, contrasts = "pairwise")

# An a priori contrast: the two active CRA arms (averaged) versus
# standard care. With levels ordered Standard, CRA, CRA + Disulfiram,
# the weights c(-1, 0.5, 0.5) compare the active arms against Standard.
contrast_test(
  fit_drinks,
  contrasts = list("CRA arms vs Standard" = c(-1, 0.5, 0.5))
)
```

`convert_cor_cov` *Correlation Matrix to Covariance Matrix Conversion*

Description

Rescales a correlation matrix into the covariance matrix implied by a set of standard deviations, the inverse of the standardization that produces a correlation matrix from a covariance matrix. Useful when a published article reports correlations and standard deviations but an analysis needs the covariances.

Usage

```
convert_cor_cov(cor_mat, sd, discrepancy = 1e-05)
```

Arguments

`cor_mat`	The correlation matrix to be converted
`sd`	A vector that contains the standard deviations of the variables in the correlation matrix
`discrepancy`	A small nonnegative tolerance (near 0; default `1e-5`). A value on the main diagonal of the correlation matrix is treated as equal to 1 when it is within `discrepancy` of 1, that is, when $\|d - 1\| \leq$ `discrepancy`

Details

The correlation matrix to convert can be either symmetric or triangular. The covariance matrix returned is always a symmetric matrix.

Value

A square numeric matrix giving the covariance matrix implied by the supplied correlation matrix and standard deviations, with the same row / column names as `cor_mat`.

Note

The correlation matrix input should be a square matrix, and the length of `sd` should be equal to the number of variables in the correlation matrix (i.e., the number of rows/columns). Sometimes the correlation matrix input may not have exactly 1's on the main diagonal, due to, e.g., rounding; `discrepancy` specifies the allowable discrepancy so that the function still considers the input as a correlation matrix and can proceed (but the function does not change the numbers on the main diagonal).

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

Other parameterization conversions: `convert_F_chisq()`, `convert_R2`, `convert_Z_r()`, `convert_d_or()`, `convert_d_r()`, `convert_r_Z()`, `convert_t_smd`, `convert_z_normal()`

Examples

```
Cor.Mat <- rbind(c(1.0000, 0.8254, 0.4261, 0.6237, 0.5901, 0.1564, 0.1551),
              c(0.8254, 1.0000, 0.5583, 0.5967, 0.6692, 0.1877, 0.2246),
              c(0.4261, 0.5583, 1.0000, 0.4933, 0.4455, 0.1472, 0.3433),
              c(0.6237, 0.5967, 0.4933, 1.0000, 0.6403, 0.1160, 0.5316),
              c(0.5901, 0.6692, 0.4455, 0.6403, 1.0000, 0.3769, 0.5742),
              c(0.1564, 0.1877, 0.1472, 0.1160, 0.3769, 1.0000, 0.2833),
              c(0.1551, 0.2246, 0.3433, 0.5316, 0.5742, 0.2833, 1.0000))
colnames(Cor.Mat) <- rownames(Cor.Mat) <- c("rating", "complaints", "privileges",
"learning", "raises", "critical", "advance")

SDs <- c(12.172562, 13.314757, 12.235430, 11.737013, 10.397226, 9.894908, 10.288706)
convert_cor_cov(cor_mat=Cor.Mat, sd=SDs)
```

`convert_d_or` *Convert Between the Standardized Mean Difference and the Odds Ratio*

Description

Invertible conversions between a two-group standardized mean difference (Cohen's d) and an odds ratio, by the logistic-distribution method of Hasselblad and Hedges (1995): a continuous outcome split at a threshold under logistic errors implies

$$d = \log(\mathrm{OR}) \cdot \frac{\sqrt{3}}{\pi}, \qquad \mathrm{OR} = \exp\!\left(d \cdot \pi / \sqrt{3}\right).$$

These conversions let binary-outcome studies enter a synthesis on the standardized mean difference scale, or mean-difference studies enter one on the odds ratio scale (Borenstein, Hedges, Higgins, & Rothstein, 2009, Chapter 7).

Usage

```
convert_d_or(d)

convert_or_d(or)
```

Arguments

`d`	The standardized mean difference.
`or`	The odds ratio, a single positive number.

Value

A `data.frame` (class `dmar_tbl`) with a single row: term `odds_ratio` (for `convert_d_or`) or `smd` (for `convert_or_d`) and its `value`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Borenstein, M., Hedges, L. V., Higgins, J. P. T., & Rothstein, H. R. (2009). *Introduction to meta-analysis*. Wiley.

Hasselblad, V., & Hedges, L. V. (1995). Meta-analysis of screening and diagnostic tests. *Psychological Bulletin, 117*(1), 167–178. doi:10.1037/00332909.117.1.167

See Also

`convert_d_r` / `convert_r_d` for the correlation leg of the same triangle.

Other parameterization conversions: `convert_F_chisq()`, `convert_R2`, `convert_Z_r()`, `convert_cor_cov()`, `convert_d_r()`, `convert_r_Z()`, `convert_t_smd`, `convert_z_normal()`

Examples

```
# d = 0.5 corresponds to an odds ratio of about 2.48.
convert_d_or(d = 0.5)

# And back, exactly.
convert_or_d(or = convert_d_or(d = 0.5)$value)

# The null maps to the null: d = 0 is an odds ratio of 1.
convert_d_or(d = 0)
```

`convert_d_r` *Convert Between the Standardized Mean Difference and the Correlation*

Description

Invertible conversions between a two-group standardized mean difference (Cohen's d) and the (point-biserial) correlation between the outcome and group membership. `convert_d_r()` maps d to r; `convert_r_d()` maps r back to d. These are the standard conversions used to bring effect sizes reported in different metrics onto a common scale, for example when synthesizing a literature in which some studies report mean differences and others report correlations (Borenstein, Hedges, Higgins, & Rothstein, 2009, Chapter 7).

Usage

```
convert_d_r(d, n_1 = NULL, n_2 = NULL)

convert_r_d(r, n_1 = NULL, n_2 = NULL)
```

Arguments

`d`	The standardized mean difference.
`n_1`, `n_2`	Optional per-group sample sizes. When supplied, the conversion uses the unequal-group factor $a = (n_1 + n_2)^2/(n_1 n_2)$; when omitted, equal group sizes are assumed, for which $a = 4$.
`r`	The point-biserial correlation, in $(-1, 1)$.

Details

With $a = (n_1 + n_2)^2/(n_1 n_2)$ (equal to 4 for equal groups), the two directions are

$$r = \frac{d}{\sqrt{d^2 + a}}, \qquad d = \frac{\sqrt{a}\, r}{\sqrt{1 - r^2}},$$

exact inverses of one another for a given a. The same `n_1` and `n_2` must be supplied to both directions for the round trip to be exact.

Value

A data.frame (class dmar_tbl) with a single row: term r (for convert_d_r) or smd (for convert_r_d) and its value.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Borenstein, M., Hedges, L. V., Higgins, J. P. T., & Rothstein, H. R. (2009). *Introduction to meta-analysis*. Wiley.

See Also

convert_d_or / convert_or_d for the odds ratio leg of the same triangle; smd and ci_r for estimating the quantities being converted.

Other parameterization conversions: convert_F_chisq(), convert_R2, convert_Z_r(), convert_cor_cov(), convert_d_or(), convert_r_Z(), convert_t_smd, convert_z_normal()

Examples

```
# Equal groups: d = 0.5 corresponds to r about .243.
convert_d_r(d = 0.5)

# And back, exactly.
convert_r_d(r = convert_d_r(d = 0.5)$value)

# Unequal groups change the conversion factor.
convert_d_r(d = 0.5, n_1 = 20, n_2 = 80)
```

---

convert_F_chisq *Convert Between an* F *Value and a Chi Square Value*

Description

Converts an observed *F* value into a chi square value, and a chi square value into an *F* value. Both functions return the converted statistic itself, a single number, not a *p*-value.

Two conversions are available, selected by df_denominator.

- **Scaling (the default).** With df_denominator = Inf, convert_F_chisq() returns df_numerator * F_value and convert_chisq_F() returns chi_square / df. This is the standard conversion between the two test statistics and needs no denominator degrees of freedom.
- **Probability matching.** With a finite df_denominator, convert_F_chisq() returns the chi square value that has the same upper-tail probability (the same *p*-value) as the *F* value, and convert_chisq_F() returns the *F* value with the same upper-tail probability as the chi square value.

Usage

```
convert_F_chisq(F_value, df_numerator, df_denominator = Inf)

convert_chisq_F(chi_square, df, df_denominator = Inf)
```

Arguments

`F_value`	Observed F value. Must be nonnegative.
`df_numerator`	Numerator degrees of freedom of the F, which is also the degrees of freedom of the chi square.
`df_denominator`	Denominator degrees of freedom of the F, the degrees of freedom on which the error variance is estimated. The default, `Inf`, treats the error variance as known and gives the scaling conversion ($\chi^2 = \nu_1 F$); a finite value gives the probability-matching conversion. See *Details*.
`chi_square`	Observed chi square value. Must be nonnegative.
`df`	Degrees of freedom of the chi square, which becomes the numerator degrees of freedom of the F.

Details

Why there are two conversions. An F statistic is the ratio of two independent chi squares, each divided by its degrees of freedom,

$$F(\nu_1, \nu_2) = \frac{\chi^2_{\nu_1}/\nu_1}{\chi^2_{\nu_2}/\nu_2},$$

where the denominator is the estimated error variance scaled to have a mean of 1. A chi square is what the numerator becomes when that error variance is known rather than estimated. This is the entire difference between the two, and it is why the conversion depends on how the error variance is treated.

Scaling: `df_denominator = Inf`**.** As the denominator degrees of freedom grow, the estimated error variance converges to the true one and $\nu_1 F \to \chi^2(\nu_1)$. The default therefore treats the error variance as known and returns exactly

$$\chi^2 = \nu_1 \, F, \qquad F = \chi^2/\nu_1,$$

with ν_1 the numerator degrees of freedom (`df_numerator`, which is also the degrees of freedom of the chi square). This is the value an F table prints in its infinite-denominator row, and it is the usual conversion between a Wald F and a Wald chi square. It involves no probabilities and needs no `df_denominator`.

Probability matching: finite `df_denominator`**.** When the error variance is estimated on ν_2 degrees of freedom, the scaling above runs high, because $\nu_1 F$ is more dispersed than $\chi^2(\nu_1)$. Supplying `df_denominator` returns instead the chi square value at the same upper-tail probability. Writing `pf` and `qchisq` for R's distribution and quantile functions, the computation is exactly

```
p <- pf(F_value, df_numerator, df_denominator, lower.tail = FALSE)
chi_square <- qchisq(p, df_numerator, lower.tail = FALSE)
```

and `convert_chisq_F()` composes the same two functions in the other order. The upper tail is used so the *p*-value is represented accurately for large statistics (the lower-tail probability rounds to 1 in double precision by about $F = 500$ at small ν_2, which would send `qchisq()` to infinity; the upper-tail value stays accurate past $F = 10^{20}$). No logarithms are involved and the returned value is the chi square itself, not the *p*-value used to find it. The map is strictly increasing and therefore one to one, and `convert_chisq_F()` is its exact inverse.

How much the two differ. At $\nu_1 = 3$ and $F = 2.75$, probability matching gives $\chi^2 = 6.290$ at $\nu_2 = 10$, 7.711 at $\nu_2 = 50$, and 8.220 at $\nu_2 = 1000$, approaching the scaling value $\nu_1 F = 8.25$ only as ν_2 grows. Scaling and probability matching agree in the limit and diverge as ν_2 shrinks; for a small ν_2, the scaled value understates the *p*-value (with $\nu_1 = 3, \nu_2 = 5$, a result whose true *p* is .05 reads as .001 if $\nu_1 F$ is referred to χ^2).

Noncentrality. Both conversions are defined by the central distributions and preserve the *p*-value; neither transports a noncentrality parameter (a noncentral *F* is not carried to a noncentral chi square with the same λ, except in the $\nu_2 \to \infty$ limit). For noncentral work use `ci_nc_F` and `ci_nc_chisq`.

Special case. With $\nu_1 = 1$ this is the squared form of the relation between *t* and *z*, since $F(1, \nu) = t(\nu)^2$ and $\chi^2(1) = z^2$.

Value

A 1-row `data.frame` with columns `term` and `value`. The `term` is `"chi_square_from_F"` for `convert_F_chisq` and `"F_from_chi_square"` for `convert_chisq_F`, and `value` is the converted statistic (a chi square value or an *F* value, respectively; never a *p*-value).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous univariate distributions* (2nd ed., Vol. 2). Wiley.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

`cv_f`, `cv_chisq`, `ci_nc_F`, `ci_nc_chisq`

Other parameterization conversions: `convert_R2`, `convert_Z_r()`, `convert_cor_cov()`, `convert_d_or()`, `convert_d_r()`, `convert_r_Z()`, `convert_t_smd`, `convert_z_normal()`

Examples

```
# Scaling (the default): chi square = df_numerator * F.
convert_F_chisq(2.75, df_numerator = 3)

# The inverse: F = chi square / df.
convert_chisq_F(8.25, df = 3)

# Probability matching at 3 and 50 degrees of freedom: the chi square
```

```
# with the same p-value as the F.
convert_F_chisq(2.75, df_numerator = 3, df_denominator = 50)

# The p-value is preserved, which is what probability matching means.
f <- 2.75
x <- convert_F_chisq(f, df_numerator = 3, df_denominator = 50)$value
c(p_from_F = pf(f, 3, 50, lower.tail = FALSE),
  p_from_chi_square = pchisq(x, 3, lower.tail = FALSE))
```

`convert_r_Z` *Convert a Correlation Coefficient (*r*) Into the Scale of Fisher's* Z

Description

This function converts a correlation coefficient into the scale of Fisher's *Z*, the variance-stabilizing transformation of a correlation. Many authors call this map the z-prime transform and write the transformed value as z'. The capital *Z* is meaningful: Fisher's *Z* is not a z-score (it is not a standardized variate, that is, an observation centered and divided by a standard deviation). It is the transform $Z = \operatorname{atanh}(r)$ of a correlation coefficient, applied because the sampling distribution of *Z* is approximately normal with a variance that does not depend on the population correlation, which makes *Z* convenient for forming confidence intervals.

Usage

```
convert_r_Z(r)
```

Arguments

`r`	Correlation coefficient (between two variables)

Details

This function is typically used in the context of forming a confidence interval for a population correlation coefficient. Note that, in that situation, the two variables are assumed to follow a bivariate normal distribution (e.g., Hays, 1994).

Value

A 1-row `data.frame` with columns `term` and `value`. The `term` is `"Z_from_r"` and `value` is Fisher's *Z* corresponding to the supplied correlation coefficient. The inverse direction is `convert_Z_r`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Hays, W. L. (1994). *Statistics* (5th ed.). Fort Worth, TX: Harcourt Brace College Publishers.

See Also

convert_Z_r, ci_r

Other parameterization conversions: convert_F_chisq(), convert_R2, convert_Z_r(), convert_cor_cov(), convert_d_or(), convert_d_r(), convert_t_smd, convert_z_normal()

Examples

```
# From Hays (1994, pp. 649--650)
convert_r_Z(.35)
```

---

convert_R2 *Convert Between* F, R^2, *and Their Noncentral Parameters*

Description

Given values of test statistics (and the appropriate additional information) the value of the noncentral values can be obtained. Likewise, given noncentral values (and the appropriate additional information) the value of the test statistic can be obtained.

Usage

```
convert_R2_f(R2 = NULL, df_1 = NULL, df_2 = NULL, p = NULL, N = NULL)

convert_f_R2(F_value = NULL, df_1 = NULL, df_2 = NULL)

convert_lambda_R2(lambda = NULL, N = NULL)

convert_R2_lambda(R2 = NULL, N = NULL)
```

Arguments

R2	Squared multiple correlation coefficient (population or observed)
df_1	Degrees of freedom for the numerator of the *F*-distribution
df_2	Degrees of freedom for the denominator of the *F*-distribution
p	Number of predictor variables for R2
N	Sample size
F_value	The obtained *F* value from a test of significance for the squared multiple correlation coefficient
lambda	The noncentral parameter from an *F*-distribution

Details

These functions are especially helpful in the search for confidence intervals for noncentral parameters, as they convert to and from related quantities.

Value

Each of the four functions returns a 1-row `data.frame` with columns `term` and `value`. The `term` entry identifies the conversion performed (`"r2_f"`, `"f_r2"`, `"lambda_r2"`, or `"r2_lambda"`) and `value` is the converted scalar. The conversions are exact inverses of one another (with the appropriate degrees-of-freedom / sample size inputs supplied), which is what makes them useful inside the noncentrality-parameter confidence interval machinery of `ci_R2` and `ci_nc_F`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

See Also

`ss_aipe_R2`, `ci_R2`, `ci_nc_t`, `ci_nc_F`

Other parameterization conversions: `convert_F_chisq()`, `convert_Z_r()`, `convert_cor_cov()`, `convert_d_or()`, `convert_d_r()`, `convert_r_Z()`, `convert_t_smd`, `convert_z_normal()`

Examples

```
convert_R2_lambda(R2 = .5, N = 100)
```

`convert_t_smd` *Conversion Functions for Noncentral* t*-distribution*

Description

Functions useful for converting a standardized mean difference to a noncentrality parameter, and vice versa.

Usage

```
convert_delta_lambda(delta, n_1, n_2)

convert_lambda_delta(lambda, n_1, n_2)
```

Arguments

delta	Population value of the standardized mean difference
n_1	Sample size in group 1
n_2	Sample size in group 2
lambda	noncentral value from a *t*-distribution

Details

Although lambda is the population noncentral value, an estimate of it is the observed value of a *t*-statistic. Likewise, delta can be estimated as the observed standardized mean difference. Thus, the observed standardized mean difference can be converted to the observed *t*-value. These functions are especially helpful in the context of forming confidence intervals for the population standardized mean difference.

Value

Each function returns a 1-row data.frame with columns term and value. The term entry identifies the conversion ("delta_lambda" or "lambda_delta") and value is the converted scalar. The two functions are exact inverses given the *per-group* sample sizes n_1 and n_2.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

See Also

smd, ci_smd, ss_aipe_smd

Other parameterization conversions: convert_F_chisq(), convert_R2, convert_Z_r(), convert_cor_cov(), convert_d_or(), convert_d_r(), convert_r_Z(), convert_z_normal()

Examples

```
convert_lambda_delta(lambda = 2, n_1 = 113, n_2 = 113)
convert_delta_lambda(delta = .266076, n_1 = 113, n_2 = 113)
```

convert_z_normal *Convert a Standard Normal* z *Value to the Corresponding Value on a Normal Distribution*

Description

This function maps a value on the standard normal distribution (the z-distribution, with mean 0 and variance 1) to the equivalent point on a normal distribution with arbitrary mean and standard deviation, $N(mean, sd^2)$.

Usage

```
convert_z_normal(z, mean = 0, sd = 1)
```

Arguments

z	A value on the standard normal distribution (with mean 0 and variance 1).
mean	The mean of the target normal distribution.
sd	The standard deviation of the target normal distribution.

Details

The conversion is `value = mean + z * sd`, which places the returned value at the same percentile of $N(mean, sd^2)$ that z occupies on the standard normal distribution. Equivalently, `value = qnorm(pnorm(z), mean, sd)`. With the defaults (`mean = 0`, `sd = 1`) the value is returned unchanged, since the target distribution is then the standard normal distribution itself.

Value

A 1-row `data.frame` with columns `term` and `value`. The `term` is `"value_from_z"` and `value` is the point on $N(mean, sd^2)$ that lies at the same percentile as `z` does on the standard normal distribution.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

cv_z

Other parameterization conversions: convert_F_chisq(), convert_R2, convert_Z_r(), convert_cor_cov(), convert_d_or(), convert_d_r(), convert_r_Z(), convert_t_smd

Examples

```
# A z value of 1.96 on the standard normal distribution maps to the
# corresponding point on a normal distribution with mean 100 and sd 15.
convert_z_normal(z = 1.96, mean = 100, sd = 15)

# With the default standard normal target, the value is returned unchanged.
convert_z_normal(z = 1.96)
```

`convert_Z_r` *Convert Fisher's* Z *Into the Scale of a Correlation Coefficient (*r*)*

Description

Converts Fisher's Z back into the scale of a correlation coefficient (r). Fisher's Z is the variance-stabilizing transformation of a correlation; many authors call it the z-prime transform and write the transformed value as z'. The capital Z is meaningful: Fisher's Z is not a z-score (it is not a standardized variate, that is, an observation centered and divided by a standard deviation). This function applies the inverse transform $r = \tanh(Z)$ to return to the scale of a correlation coefficient.

Usage

```
convert_Z_r(Z)
```

Arguments

`Z`	Fisher's Z (the variance-stabilizing transform of a correlation, which many authors call z')

Details

This function is typically used in the context of forming a confidence interval for a population correlation coefficient. Note that, in that situation, the two variables are assumed to follow a bivariate normal distribution (e.g., Hays, 1994).

Value

A 1-row `data.frame` with columns `term` and `value`. The `term` is `"r_from_Z"` and `value` is the correlation coefficient corresponding to the supplied Fisher's Z. The inverse direction is `convert_r_Z`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Hays, W. L. (1994). *Statistics* (5th ed.). Fort Worth, TX: Harcourt Brace College Publishers.

See Also

convert_r_Z, ci_r

Other parameterization conversions: convert_F_chisq(), convert_R2, convert_cor_cov(), convert_d_or(), convert_d_r(), convert_r_Z(), convert_t_smd, convert_z_normal()

Examples

```
# From Hays (1994, pp. 649--650)
convert_Z_r(0.3654438)
```

`correction_for_attenuation`

Correct a Correlation for Attenuation Due to Measurement Error

---

Description

Applies the Spearman (1904) correction for attenuation: the observed correlation between two fallible measures understates the correlation between the constructs they measure, and dividing by the square root of the product of the two reliabilities recovers it,

$$r_c = \frac{r_{XY}}{\sqrt{\rho_{XX'}\,\rho_{YY'}}}.$$

Within classical test theory (Lord & Novick, 1968), the disattenuated correlation estimates the correlation between the true scores, that is, how strongly the two constructs would correlate if each were measured without error. When `N` is supplied, a confidence interval for the corrected correlation is formed by disattenuating the endpoints of the Fisher's Z interval for the observed correlation, the standard practice when the reliabilities are treated as known.

Usage

```
correction_for_attenuation(
  r,
  reliability_x,
  reliability_y,
  N = NULL,
  conf_level = 0.95
)
```

Arguments

`r`	The observed correlation between the two measures, in $[-1, 1]$.
`reliability_x`	Reliability of the first measure, in $(0, 1]$. Any reliability estimate appropriate to the use may be supplied (for example, coefficient alpha or omega from reliability).

`reliability_y` Reliability of the second measure, in $(0, 1]$. For a correlation between a measure and an error-free criterion, supply 1 for that side (correcting for criterion unreliability only yields what the validity generalization literature calls the operational validity).

`N` Optional sample size on which `r` is based. When supplied, a `conf_level` confidence interval for the corrected correlation is reported by correcting the endpoints of the Fisher's Z interval for `r`.

`conf_level` Confidence level for the interval when `N` is supplied. Defaults to 0.95.

Details

The correction treats the two reliabilities as known constants, which is the conventional assumption; uncertainty in the reliabilities themselves would widen the interval further. Because the observed correlation can exceed what the supplied reliabilities allow (sampling error, or reliabilities that understate the truth), the corrected value can exceed 1 in magnitude; when that happens the value is reported as computed, with a warning, rather than silently truncated, since a corrected correlation beyond 1 is itself diagnostic information about the inputs.

Prefer the factor model when you have the items. The Spearman formula is the summary-statistics route: it is exactly right when all you have are the observed correlation and reliability estimates. When the item-level data are available, the better practice is to estimate the construct-level correlation directly as the factor correlation in a two-factor model (each scale loading on its own factor, factors free to correlate): the latent correlation is then estimated jointly with the measurement model rather than assembled from plug-in reliabilities, and it comes with a standard error that propagates the sampling variability of all the moving parts. The example below shows both routes on the same data, the formula route using `reliability_omega`, the model route using **lavaan**; with congeneric items the two agree closely, and when they disagree the factor model is the one to trust.

Value

A `data.frame` (class `dmar_tbl`) in `term` / `value` layout with the observed correlation (`correlation_observed`), the corrected correlation (`correlation_corrected`), the corrected interval (`lower_limit`, `upper_limit`; present only when `N` is supplied), and the `reliability_x`, `reliability_y`, and `N` inputs.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Lord, F. M., & Novick, M. R. (1968). *Statistical theories of mental test scores*. Addison-Wesley.

Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology, 15*(1), 72–101.

See Also

`reliability` and its family for estimating the reliabilities supplied here; `cfa_1` and **lavaan** for the latent variable route the Details recommend when items are available; `ci_r` for inference on

the observed correlation itself; convert_r_Z and convert_Z_r for the Fisher transformation the interval uses.

Other effect size estimates: cles(), cliff_delta(), eta_squared(), eta_squared_generalized(), eta_squared_partial(), expected_partial_r(), expected_r(), expected_smd(), nnt_from_smd(), omega_squared(), omega_squared_partial(), probability_of_superiority_paired(), proportion_of_superiorit responder_analysis(), smd_trimmed()

Examples

```
# An observed correlation of .30 between measures with reliabilities .80
# and .70 corresponds to a construct-level correlation of about .40.
correction_for_attenuation(r = 0.30, reliability_x = 0.80, reliability_y = 0.70)

# With the sample size, the corrected interval comes along.
correction_for_attenuation(r = 0.30, reliability_x = 0.80, reliability_y = 0.70,
                    N = 120)

# Correct one side only (error-free criterion).
correction_for_attenuation(r = 0.30, reliability_x = 0.80, reliability_y = 1)

# The two routes to the construct-level correlation, on the same data
# (requires lavaan). Two congeneric scales of three items each whose
# latent variables correlate .50:
set.seed(113)
n <- 400
fx <- rnorm(n); fy <- 0.5 * fx + sqrt(1 - 0.25) * rnorm(n)
lam <- c(.8, .7, .6)
items <- data.frame(
  x1 = lam[1] * fx + rnorm(n, 0, sqrt(1 - lam[1]^2)),
  x2 = lam[2] * fx + rnorm(n, 0, sqrt(1 - lam[2]^2)),
  x3 = lam[3] * fx + rnorm(n, 0, sqrt(1 - lam[3]^2)),
  y1 = lam[1] * fy + rnorm(n, 0, sqrt(1 - lam[1]^2)),
  y2 = lam[2] * fy + rnorm(n, 0, sqrt(1 - lam[2]^2)),
  y3 = lam[3] * fy + rnorm(n, 0, sqrt(1 - lam[3]^2)))

# Route 1, summary statistics: omega reliabilities into the formula.
x_score <- rowMeans(items[, 1:3]); y_score <- rowMeans(items[, 4:6])
om_x <- reliability_omega(data = items[, 1:3])$value[1]
om_y <- reliability_omega(data = items[, 4:6])$value[1]
correction_for_attenuation(r = cor(x_score, y_score),
                           reliability_x = om_x, reliability_y = om_y,
                           N = n)

# Route 2, the factor model: the latent correlation estimated directly.
fit <- lavaan::cfa("X =~ x1 + x2 + x3\nY =~ y1 + y2 + y3",
                   data = items, std.lv = TRUE)
lavaan::parameterEstimates(fit)[
  lavaan::parameterEstimates(fit)$op == "~~" &
  lavaan::parameterEstimates(fit)$lhs == "X" &
  lavaan::parameterEstimates(fit)$rhs == "Y", ]
```

`correlations_test` *Formatted Correlation Matrix With* p*-values and Confidence Intervals*

Description

Computes a correlation matrix along with, for every pair of variables, the two-sided *p*-value, a confidence interval, and the pairwise sample size, and arranges them into a single annotated table. The table is a convenience for inspecting the correlations, their significance, and their intervals at a glance; it is not intended as a finished, publication-ready exhibit. Output formats include plain text (for the console or to paste into a Word document), HTML (best for Word via browser copy-paste), and LaTeX.

Usage

```
correlations_test(
  x,
  method = "pearson",
  conf_level = 0.95,
  listwise = FALSE,
  stars = FALSE,
  decimals_r = 2,
  decimals_p = 4,
  format = "text",
  file = NULL
)
```

Arguments

`x`	A `data.frame`, tibble, or `matrix`. Non-numeric columns are dropped with no warning; the remaining numeric, integer, and logical columns are used.
`method`	The correlation method: `"pearson"` (default), `"spearman"`, or `"kendall"`.
`conf_level`	Confidence level for the interval (default `0.95`).
`listwise`	Logical. If `TRUE`, apply listwise deletion before any correlation is computed so that every pair uses the same sample. If `FALSE` (the default), each pair uses all of its available complete observations ("pairwise" deletion).
`stars`	Logical. If `TRUE`, append conventional significance stars (* $p < .05$, ** $p < .01$, *** $p < .001$) to each correlation and add an explanatory footnote to the table. The exact *p*-value is always shown as well.
`decimals_r`	Number of decimals for correlations and confidence interval limits (default `2`).
`decimals_p`	Number of decimals for *p*-values (default `4`, matching the package-wide `digits_p` convention used by `dmar_tbl`); values below $10^{-\text{decimals\_p}}$ are printed as "< .0001" (with the threshold tracking `decimals_p`).
`format`	One of `"text"` (default), `"html"`, or `"latex"`. Controls the returned/printed table. See Details.

`file` Optional file path. If supplied, the formatted table is written to this file. The HTML path writes a self-contained HTML document (the kable wrapped in a minimal document head that pulls in Bootstrap CSS from a CDN) so the file opens directly in a browser without any pandoc / webshot machinery. The LaTeX path writes the raw `tabular` fragment; embed it in a document that loads `\usepackage{makecell}` and `\usepackage{booktabs}`. The text path uses `writeLines`.

Details

Layout. Each lower-triangle cell stacks four values: the correlation (with optional significance stars), the two-sided *p*-value, the confidence interval, and the pairwise sample size. The upper triangle is left blank so the same information is not repeated.

***p*-values.** Computed with `cor.test` using the requested `method`. For Spearman and Kendall with ties, `cor.test` cannot compute an exact *p*-value and falls back to a normal-approximation *p*-value; the associated warnings are suppressed for a cleaner table.

Confidence intervals. All three methods use Fisher's variance- stabilizing transformation, $z(r) = \operatorname{atanh}(r)$, and back-transform through $\tanh(\cdot)$, but the standard error in the Fisher-z scale is selected to match the sampling distribution of the chosen correlation coefficient:

- Pearson (`method = "pearson"`): $\mathrm{SE}(z) = 1/\sqrt{n-3}$. This is the classical Fisher (1921) interval. Under bivariate normality the coverage of this interval matches `cor.test`'s `conf.int` exactly; for departures from bivariate normality both intervals lose coverage in the same way. See Kelley (2007) and Maxwell, Delaney, & Kelley (2027, Chapter 9) for discussion and worked examples.
- Spearman (`method = "spearman"`): $\mathrm{SE}(z) = \sqrt{(1 + r^2/2)/(n-3)}$. This is the Bonett and Wright (2000) adjustment, which uses Fisher's transformation but inflates the standard error to account for the heavier-than-Pearson tails of the Spearman sampling distribution. This is the form Bonett and Wright recommend for practical use; a plain Fisher $1/\sqrt{n-3}$ standard error tends to produce intervals that are too narrow for Spearman correlations.
- Kendall (`method = "kendall"`): $\mathrm{SE}(z) = \sqrt{0.437/(n-4)}$. This is Bonett and Wright's (2000, equation 2) Fisher-z interval for Kendall's τ. The constant 0.437 is the asymptotic variance factor of Fieller, Hartley, and Pearson (1957), derived under bivariate normality and stated by Bonett and Wright as accurate for $|\tau| < .8$ (the Spearman variance above is likewise stated as accurate for $|\rho_s| < .95$). Requires $n \geq 5$; for smaller pairwise samples the interval is returned as `NA`.

The Fisher-z machinery requires $|r| < 1$ for the transformation to be finite. When $r = \pm 1$ (perfect correlation in the sample), the transformed value is infinite and the interval is reported as `NA`; this is the same convention used by `cor.test`.

When to use each correlation. Pick `method = "pearson"` when both variables are continuous, approximately linearly related, and roughly bivariate normal (or at least without heavy tails and influential outliers). Pick `method = "spearman"` or `method = "kendall"` when the relationship is monotone but not necessarily linear, when one or both variables are ordinal, or when influential outliers would distort Pearson's *r*. Kendall's τ is often preferred over Spearman's ρ for small samples and for samples with many tied ranks because it has better small-sample properties and a more interpretable concordance-based meaning. See Maxwell, Delaney, & Kelley (2027, Chapter 9) for an extended discussion of effect size choice and interval estimation.

HTML/LaTeX output. Built with `knitr::kable` (a `Suggests` dependency). Cell content and variable names are escaped for the target format so that "$p < .001$" renders correctly and that variable names containing characters such as `_`, `&`, or `%` do not break LaTeX compilation. LaTeX output uses `\makecell`, which requires `\usepackage{makecell}` in the document preamble.

Pasting into Word. The cleanest path is `format = "html"` with a `file` argument; the function writes a small self-contained HTML document (no pandoc dependency). Open the result in a browser and copy/paste the table into Word. Formatting (including the stacked-cell layout) is preserved.

Value

An object of class `"correlations_test"` containing the matrices `r`, `p`, `ci_lower`, `ci_upper`, and `n` (all $p \times p$ with variable names as row/column names), plus the arguments used. When `format = "html"` or `format = "latex"` and `file` is `NULL`, a `kable` object is returned instead so that the table renders inside R Markdown / Quarto documents. When `format = "text"`, the table is printed to the console and the raw object is returned invisibly.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bonett, D. G., & Wright, T. A. (2000). Sample size requirements for estimating Pearson, Kendall and Spearman correlations. *Psychometrika, 65*(1), 23–28. doi:10.1007/BF02294183

Fieller, E. C., Hartley, H. O., & Pearson, E. S. (1957). Tests for rank correlation coefficients. I. *Biometrika, 44*(3/4), 470–481. doi:10.1093/biomet/44.34.470

Fisher, R. A. (1915). Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika, 10*(4), 507–521. doi:10.1093/biomet/10.4.507

Fisher, R. A. (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron, 1*, 3–32.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

`descriptives`, `cor.test`, `cor`, `ci_r`

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `equivalence_r()`, `equivalence_smd()`, `factorial_anova()`, `manova_split_plot()`, `mauchly_test()`, `mixed_anova()`, `obrien_test()`, `pairwise_within()`, `randomization_test()`, `randomization_test_paired()`, `regions_of_significance()`, `simple_effects_AB()`, `summary_t_test()`, `welch_t()`

Examples

```
# Worked example using four cognitive tests from the Holzinger and
# Swineford (1939) study (301 children in two schools). The goal of
# correlations_test() is to produce a formatted correlation matrix that
# reports, for every variable pair, the correlation, its two-sided
# p-value, a confidence interval on the population correlation, and the
# pairwise sample size. See Kelley (2007) and Maxwell, Delaney, & Kelley
# (2027, Chapter 9) for discussion of why effect sizes should be
# accompanied by confidence intervals.
hs_tests <- holzinger_swineford[, c("t1_visual_perception", "t2_cubes",
                                    "t4_lozenges",
                                    "t6_paragraph_comprehension")]

# Pearson correlations (the default). Each lower-triangle cell stacks r,
# the two-sided p-value, the 95\% confidence interval (Fisher's Z
# transformation; Fisher, 1915, 1921), and the pairwise N.
correlations_test(hs_tests)

# Add significance stars and an explanatory footnote.
correlations_test(hs_tests, stars = TRUE)

# Spearman correlations at a 99\% confidence level. The interval uses
# Bonett and Wright's (2000) Fisher's Z standard error
# sqrt((1 + r^2/2) / (n - 3)), which corrects the plain Fisher interval
# for the heavier tails of Spearman's sampling distribution.
correlations_test(hs_tests, method = "spearman", conf_level = 0.99)

# Kendall's tau, also using Bonett and Wright's (2000) Fisher's Z standard
# error sqrt(0.437 / (n - 4)). Kendall is often preferred over Spearman
# for small samples and for samples with many tied ranks, and these
# integer test scores carry many ties.
correlations_test(hs_tests, method = "kendall")

# Save a formatted HTML table that opens directly in a browser
# (then copy into Word). No pandoc required.
tmp_html <- tempfile(fileext = ".html")
correlations_test(hs_tests, stars = TRUE, format = "html", file = tmp_html)
```

---

cov_sem *Model Implied Covariance Matrix From a Lavaan-Specified SEM*

Description

Given a structural equation model written in lavaan model syntax with all of its parameters fixed to their population values, compute the model implied population covariance matrix $\Sigma(\theta)$ of the observed variables and, when the model has a mean structure, the model implied population mean vector $\mu(\theta)$. This function requires **lavaan** to be installed.

This is the helper that drives the *population* side of the sample size planning workflow for SEM: it lets the user state a population model, obtain the $\Sigma(\theta)$ (and $\mu(\theta)$) those fixed values imply, and then pass that population to `ss_aipe_sem_path`, `ss_aipe_sem_path_sensitivity`, `ss_aipe_rmsea_sensitivity`, `ss_power_composite_sem`, or `ss_aipe_composite_sem`.

Usage

```
cov_sem(model)
```

Arguments

`model`	A single character string giving a structural equation model in lavaan model syntax (see `model.syntax`), with every parameter fixed to its population value. A factor loading, a structural path, a variance, or a covariance is fixed by prefixing the numeric value to the variable with the `*` operator, for example `"f1 =~ 1*y1 + 0.8*y2 + 0.8*y3"` for the loadings, `"f2 ~ 0.5*f1"` for a structural path, and `"y1 ~~ 0.5*y1"` for a residual variance. A model with a mean structure (for example a latent growth curve model) also fixes every intercept and latent mean, for example `"t1 ~ 0*1"` and `"s ~ 0.3*1"`. lavaan syntax embeds the values and knows which variables are latent, so no separate parameter vector or list of latent variables is needed.

Details

The function builds a non-fitted lavaan object from `model` with all parameters held at the population values written into the syntax, and reads back the model implied covariance matrix of the observed variables. Because the object is created with `do.fit = FALSE`, no estimation is performed and the placeholder sample covariance lavaan needs to construct the object is never used; the returned $\Sigma(\theta)$ comes entirely from the fixed parameter values. The observed-variable names are taken from the model syntax and fix the row and column order of the returned matrix.

Value

A list with components:

`sigma_theta` The model implied population covariance matrix of the observed variables, with rows and columns named.

`mu_theta` The model implied population mean vector of the observed variables, named, in the row order of `sigma_theta`. A vector of zeros when the model has no mean structure.

`observed_vars` Character vector of observed variable names in the row/column order of `sigma_theta`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Lai, K., & Kelley, K. (2011). Accuracy in parameter estimation for targeted effects in structural equation modeling: Sample size planning for narrow confidence intervals. *Psychological Methods, 16*(2), 127–148. doi:10.1037/a0021764

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48*(2), 1–36. doi:10.18637/jss.v048.i02

See Also

sem, model.syntax, ss_aipe_sem_path, ss_aipe_sem_path_sensitivity, ss_aipe_rmsea_sensitivity, covmat_from_cfa.

Examples

```
# Population model with all parameters fixed to their values: two factors,
# three indicators each, and a structural path of 0.5 from the first
# factor to the second.
pop_model <- "
  f1 =~ 1*y1 + 0.8*y2 + 0.8*y3
  f2 =~ 1*y4 + 0.8*y5 + 0.8*y6
  f2 ~ 0.5*f1
  f1 ~~ 1*f1
  f2 ~~ 0.75*f2
  y1 ~~ 0.5*y1; y2 ~~ 0.5*y2; y3 ~~ 0.5*y3
  y4 ~~ 0.5*y4; y5 ~~ 0.5*y5; y6 ~~ 0.5*y6
"
cov_sem(pop_model)$sigma_theta

# A population model with a mean structure: a linear latent growth curve
# over four waves. The intercepts and latent means are fixed too, and
# mu_theta carries the model implied wave means, which start at 5.0 and
# rise by 0.3 per wave.
pop_lgm <- "
  i =~ 1*t1 + 1*t2 + 1*t3 + 1*t4
  s =~ 0*t1 + 1*t2 + 2*t3 + 3*t4
  i ~~ 1*i
  s ~~ 0.2*s
  i ~~ -0.15*s
  t1 ~~ 0.5*t1; t2 ~~ 0.5*t2; t3 ~~ 0.5*t3; t4 ~~ 0.5*t4
  t1 ~ 0*1; t2 ~ 0*1; t3 ~ 0*1; t4 ~ 0*1
  i ~ 5*1
  s ~ 0.3*1
"
cov_sem(pop_lgm)$mu_theta
```

`covmat_from_cfa` *Generate a Population Covariance Matrix From a One-Factor Confirmatory Factor Model*

Description

Given a vector of factor loadings (λ) and the corresponding vector of unique (error) variances (ψ^2), this function computes the implied population covariance matrix under a single-factor confirmatory factor model:

$$\Sigma = \Lambda\Lambda^\top + \Psi.$$

This builds the model implied covariance for a confirmatory factor model with uncorrelated errors. The unique-variances matrix Ψ is strictly diagonal, so no residual covariances (correlated uniquenesses) are allowed; a model with correlated errors is outside the scope of this function. Because the model is a single common factor, the loadings matrix Λ reduces to a column vector λ of length p (one per indicator), and the unique-variances matrix Ψ is the diagonal $\mathrm{diag}(\psi^2)$ of a length-p vector. The formals are named in the lowercase vector form (`lambda` and `psi_squared`) to reflect this; the matrix-form symbols (Λ, Ψ) remain in the mathematical exposition above.

Usage

```
covmat_from_cfa(lambda, psi_squared, ...)
```

Arguments

`lambda`	A numeric vector of factor loadings (one per indicator). Can also be supplied as a single-row or single-column matrix and will be coerced to a vector.
`psi_squared`	A numeric vector of unique (error) variances, one per indicator. Recycled to match the length of `lambda` when a single value is supplied (equal-error-variance model).
`...`	Optional advanced controls. Currently the only recognized passthrough is `tol_det`: the tolerance below which the determinant of the implied covariance matrix triggers a positive- definite warning (default `1e-05`). The argument is hidden in `...` because the default is appropriate for almost every application; users who pass an unrecognized name through `...` (for example, a misspelling) are notified with a warning so the typo is not silently ignored.

Details

Under the single-factor common-factor model each indicator score is $x_i = \lambda_i\xi + \delta_i$, where ξ is the (standardized) latent factor and δ_i is the indicator-specific residual with variance ψ_i^2. The population covariance among the manifest indicators is therefore

$$\Sigma = \Lambda\Lambda^\top + \Psi,$$

where Λ is the $p \times 1$ column of loadings $(\lambda_1, \ldots, \lambda_p)^\top$ and $\Psi = \mathrm{diag}(\psi_1^2, \ldots, \psi_p^2)$. In code we work with the vectors `lambda` and `psi_squared` directly. Because Ψ is built with `diag()` from a length-p vector, the errors are uncorrelated by construction: there is no way to specify a residual covariance between two indicators. A confirmatory factor model with correlated errors (correlated uniquenesses) requires a more general formulation than this function provides.

The `cfa` spelling matches the `cfa_1` naming.

Value

A list with the single element `population_cov`: the implied population covariance matrix of the manifest indicators ($p \times p$, symmetric).

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

`cfa_1`, `ss_aipe_reliability`

Examples

```
# Five indicators with equal loadings and equal error variances
covmat_from_cfa(lambda = rep(0.7, 5), psi_squared = rep(0.51, 5))

# Unequal loadings
covmat_from_cfa(lambda      = c(0.5, 0.6, 0.7, 0.8),
                psi_squared = c(0.75, 0.64, 0.51, 0.36))
```

---

cv *Coefficient of Variation (Biased or Unbiased Estimator)*

Description

Computes the sample coefficient of variation $\hat{\kappa} = s/\bar{Y}$ or, optionally, its first-order bias-corrected counterpart under normality. Either supply a precomputed `cv` or supply the raw `mean` and `sd`; with `unbiased = TRUE` the value is multiplied by the small-sample correction $(1+1/(4N))$. The (biased) sample coefficient of variation (the default, `unbiased = FALSE`) is the form usually reported. To accompany it with a confidence interval, use `ci_cv`.

Usage

```
cv(cv = NULL, mean = NULL, sd = NULL, N = NULL, unbiased = FALSE)
```

Arguments

`cv`	The sample coefficient of variation, $s/\bar{Y}$. Optional; if supplied, `mean` and `sd` must not be.
`mean`	Sample mean. Numeric scalar.
`sd`	Sample standard deviation, using $N - 1$ in the variance denominator. Numeric scalar.
`N`	Sample size. Required when `unbiased = TRUE`.
`unbiased`	Logical. If `TRUE`, applies the first-order small-sample bias correction $(1+1/(4N))$ to the plug-in estimator. Default `FALSE`.

Details

The plug-in estimator $\hat{\kappa} = s/\bar{Y}$ is the workhorse coefficient of variation in applied work and is the form usually reported. It is what this function returns by default (`unbiased = FALSE`). A point estimate is most informative when paired with a confidence interval; `ci_cv` computes one for κ. Under normality, however, the plug-in estimator is biased downward, and the leading-order expansion is $E[\hat{\kappa}] = \kappa(1 - 1/(4N)) + O(N^{-2})$ (Sokal & Rohlf, 1995). The bias is negligible at N above about 100 but is non-trivial in small samples, where multiplying the plug-in value by $(1 + 1/(4N))$ removes the leading-order term. The `unbiased = TRUE` option applies that correction.

For confidence intervals on κ, the McKay (1932) noncentral t based interval is implemented in `ci_cv`; the corresponding asymptotic variances are in `var_cv`. The Vangel (1996) small-sample refinement of McKay's interval is a further option described in the literature; it matters more than the bias correction in this function when κ is larger than about 0.3 (Kelley, 2007).

Value

A 1-row `data.frame` with columns `term` and `value`. The `term` value is `"cv"` and `value` is either the plug-in estimator (default) or the first-order bias-corrected estimator (when `unbiased = TRUE`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Sample size planning for the coefficient of variation from the accuracy in parameter estimation approach. *Behavior Research Methods, 39*(4), 755–766. doi:10.3758/BF03192966

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3.)

McKay, A. T. (1932). Distribution of the coefficient of variation and the extended *t* distribution. *Journal of the Royal Statistical Society, 95*(4), 695–698.

Sokal, R. R., & Rohlf, F. J. (1995). *Biometry: The principles and practice of statistics in biological research* (3rd ed.). W. H. Freeman.

Vangel, M. G. (1996). Confidence intervals for a normal coefficient of variation. *The American Statistician, 50*(1), 21–26. doi:10.1080/00031305.1996.10473537

See Also

`ci_cv`, `var_cv`, `ss_aipe_cv`

Examples

```
# 1. Point estimate from raw mean and SD.
cv(mean = 100, sd = 15)

# 2. Bias-corrected estimate at N = 50; the correction is small but
#    non-negligible at this sample size.
cv(mean = 100, sd = 15, N = 50, unbiased = TRUE)
```

```
# 3. Bias correction at N = 10; the correction is larger here.
cv(cv = .15, N = 10, unbiased = TRUE)
```

`cv_bonferroni_f` *Provides the Bonferroni-Adjusted Critical Value for an* F *Test of One of Several Contrasts*

Description

Provides the Bonferroni-Adjusted Critical Value for an *F* Test of One of Several Contrasts

Usage

```
cv_bonferroni_f(
  alpha_level = 0.05,
  df_denominator,
  n_comparisons,
  df_numerator = 1,
  verbose = TRUE
)
```

Arguments

`alpha_level`	The family-wise Type I error rate (i.e., the rate for the set of `n_comparisons` tests taken together). Default `0.05`, the level Maxwell, Delaney, and Kelley (2027) tabulate.
`df_denominator`	The denominator (error) degrees of freedom (a positive number). In a one-way design with N observations and a groups this is $N - a$.
`n_comparisons`	The number of comparisons in the family, C (a positive integer).
`df_numerator`	The numerator degrees of freedom. Default `1`, because a contrast carries a single degree of freedom, which is the case the Appendix table covers.
`verbose`	Provides extra information about areas under the curve.

Details

The Bonferroni adjustment tests each of C contrasts at α/C rather than α, which holds the family-wise error rate at or below α whatever the contrasts are and however they are correlated. The critical value is therefore an ordinary upper-tail *F* quantile read at the smaller per-comparison rate,

$$F_{\alpha/C;\,\mathrm{df}_{\mathrm{num}},\,\mathrm{df}_{\mathrm{den}}},$$

which is what `cv_f` would return if handed `alpha / C`. This function exists because the adjustment is worth naming: the whole of it is the division, and seeing α/C reported back in `area_greater` is the point. Maxwell, Delaney, and Kelley (2027) tabulate these values for one numerator degree of freedom and a family-wise alpha of .05 in their Appendix Table A.3.

The default df_numerator = 1 covers the case the table addresses and the one that arises in practice, since a contrast among means is a single-degree-of-freedom question. Supply a larger value to Bonferroni adjust a family of multiple-degree-of-freedom model comparisons.

The procedure is often called Dunn's, after Dunn (1961), who first applied the Bonferroni inequality to multiple contrasts. It is not the rank-sum procedure of dunn_test, which the same author published three years later.

When something else is better. Bonferroni makes no use of the structure of the family, so a procedure built for a particular structure beats it there: cv_tukey_hsd is more powerful for all pairwise comparisons, and cv_dunnett is more powerful for comparing several treatments to one control. Bonferroni's advantage is generality; it applies to any set of contrasts chosen in advance, and it can beat Tukey's method when only a few of the pairwise comparisons were planned. Because it is conservative, a step-down variant such as Holm's is uniformly more powerful while controlling the same rate, and is available through the method argument of contrast_adjusted and p.adjust.

Value

Returns the critical value in a output style (a data.frame following the format used by cv_t). When verbose = TRUE the area_greater column reports the per-comparison error rate α/C that the adjustment spends on each test.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association, 56*(293), 52–64. doi:10.2307/2282330

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 5 on the multiple-comparisons problem; Appendix Table A.3 reports these critical values.)

See Also

cv_f, cv_tukey_hsd, cv_dunnett, cv_scheffe, contrast_adjusted

Other critical values: cv_bryant_paulson(), cv_chisq(), cv_dunnett(), cv_f(), cv_scheffe(), cv_smm(), cv_t(), cv_tukey_hsd(), cv_z()

Examples

```
# Five planned contrasts with 20 error degrees of freedom, holding the
# family-wise error rate at .05. The area_greater column shows the .01
# per-comparison rate the adjustment spends on each test.
cv_bonferroni_f(alpha_level = .05, df_denominator = 20, n_comparisons = 5)

# It is the F critical value read at alpha / C.
cv_bonferroni_f(alpha_level = .05, df_denominator = 20, n_comparisons = 5,
                verbose = FALSE)$value
cv_f(alpha_level = .05 / 5, df_numerator = 1, df_denominator = 20,
```

```
    verbose = FALSE)$value[2]

# With one comparison there is nothing to adjust.
cv_bonferroni_f(alpha_level = .05, df_denominator = 20, n_comparisons = 1,
                verbose = FALSE)$value
```

`cv_bryant_paulson` *Provides the Critical Value for the Bryant–Paulson ANCOVA Multiple-Comparison Procedure*

Description

Computes the critical value of the Bryant–Paulson generalized studentized range, the reference distribution for multiple comparisons of adjusted means in an analysis of covariance with random covariates. The single-step value is the multiplier for simultaneous confidence intervals on pairwise differences of adjusted means; the `"duncan"` option instead returns the significant range of the stepwise Duncan multiple-range procedure.

Usage

```
cv_bryant_paulson(
  alpha_level,
  df,
  groups,
  covariates = 1,
  procedure = c("tukey", "duncan"),
  verbose = TRUE
)
```

Arguments

`alpha_level`	Type I error rate (i.e., the false positive rate). As with `cv_tukey_hsd`, the full `alpha_level` applies to the upper tail of the (non-negative) generalized studentized range distribution; it is not split between two tails (see Details).
`df`	The ANCOVA error degrees of freedom (a positive number; in a one-way ANCOVA, `df = N - groups - covariates`).
`groups`	The number of groups whose adjusted means are being compared (an integer of at least 2).
`covariates`	The number of *random* covariates in the ANCOVA, the parameter p of the Bryant–Paulson distribution (a non-negative integer). With `covariates = 0` the critical value reduces to the ordinary studentized range and the result equals $\sqrt{2}$ times `cv_tukey_hsd` (see Details).
`procedure`	One of `"tukey"` (default) or `"duncan"`. `"tukey"` returns the single-step *simultaneous* critical value $q_{\alpha;p,k,\nu}$ used for familywise (Tukey–Kramer-type) confidence intervals; `"duncan"` returns the stepwise Duncan multiple-range significant range $r_{\alpha;p,k,\nu}$ tabled by Bryant and Bruvold (1980, Table 2) (see Details).
`verbose`	Provides extra information (the tail areas) about the critical value.

Details

The Bryant–Paulson procedure is the analysis-of-covariance generalization of Tukey's method (cv_tukey_hsd) for comparing adjusted means when the covariate is *random*. Because the covariate adjustment must be estimated, the studentized range of adjusted means is stochastically larger than the ordinary studentized range, so the Bryant–Paulson critical value exceeds Tukey's; using the latter would give intervals that are too narrow and a familywise error rate above alpha_level. The single-step (procedure = "tukey") value is the multiplier for a family of simultaneous confidence intervals on the pairwise differences of adjusted means that jointly hold at level $1 - \alpha$. Maxwell, Delaney, and Kelley (2027, Chapter 9) develop multiple comparisons of adjusted means in the analysis of covariance, the setting this critical value serves.

The reference distribution is the Bryant–Paulson generalized studentized range, implemented in qbryant_paulson. Its quantiles are not a standard base-R distribution and are not the multivariate *t* quantiles that cv_dunnett and cv_smm obtain from **mvtnorm**; they are computed directly by qbryant_paulson, so this function depends on neither base-R nor **mvtnorm** multiple-mean machinery.

Scale. The returned value is on the studentized-range scale, $q_{\alpha;p,k,\nu}$, the scale of Bryant and Paulson's (1976) and Bryant and Bruvold's (1980) tables and of Eq. (2.4) of the latter. A pair of adjusted means is declared different when $|\hat{\theta}_i - \hat{\theta}_j| > q_{\alpha;p,k,\nu}\,\hat{\sigma}_{y|x}\sqrt{1/n}$. This differs from cv_tukey_hsd, which divides its value by $\sqrt{2}$ to report on the pairwise mean-difference scale; divide the value here by $\sqrt{2}$ to obtain that scale. With covariates = 0, cv_bryant_paulson returns exactly $\sqrt{2}\times$ cv_tukey_hsd.

Duncan multiple-range. For procedure = "duncan" the value is the "significant range" of Duncan's stepwise test as extended to ANCOVA by Bryant and Bruvold (1980, Section 4). With variable protection levels $\alpha_k = 1 - (1-\alpha)^{k-1}$,

$$r_{\alpha;p,2,\nu} = q_{\alpha;p,2,\nu}, \qquad r_{\alpha;p,k,\nu} = \max\{\, r_{\alpha;p,k-1,\nu},\ q_{\alpha_k;p,k,\nu} \,\}, \quad k > 2.$$

These are the values in Bryant and Bruvold's Table 2, reproduced by this function to the tabled two-decimal precision (see the package tests).

Value

Returns the critical value in a output style (a data.frame with class dmar_tbl and one row per critical value, following the format used by cv_tukey_hsd and cv_t). The value is on the *studentized-range scale* (the scale on which Bryant and Paulson tabulate their critical values and on which ci_c_ancova_bp uses them). When verbose = TRUE and procedure = "tukey", the upper- and lower-tail areas of the Bryant–Paulson distribution at the critical value are also returned; for procedure = "duncan" the tail areas are NA because the significant range is a stepwise quantity rather than a single quantile.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bryant, J. L., & Paulson, A. S. (1976). An extension of Tukey's method of multiple comparisons to experimental designs with random concomitant variables. *Biometrika, 63*, 631–638.

Bryant, J. L., & Bruvold, N. T. (1980). Multiple comparison procedures in the analysis of covariance. *Journal of the American Statistical Association, 75*(372), 874–880. doi:10.2307/2287175

Duncan, D. B. (1955). Multiple range and multiple F tests. *Biometrics, 11*, 1–42.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9, where multiple comparisons of adjusted means in the analysis of covariance are developed; Appendix Table A.8 reports these critical values.)

See Also

cv_tukey_hsd, cv_scheffe, qbryant_paulson, ci_c_ancova_bp

Other critical values: cv_bonferroni_f(), cv_chisq(), cv_dunnett(), cv_f(), cv_scheffe(), cv_smm(), cv_t(), cv_tukey_hsd(), cv_z()

Examples

```
# Multiple comparisons of adjusted means in ANCOVA (the setting of Maxwell,
# Delaney, and Kelley, 2027, Chapter 9), using the worked example of Bryant
# and Bruvold (1980): 6 panels, 1 random covariate, 14 error df,
# alpha_level = .05. The single-step value is the multiplier for simultaneous
# confidence intervals on the pairwise differences of adjusted means. With a
# covariate present there is no closed form for it: the Bryant-Paulson
# distribution function is integrated numerically and then inverted. The
# value is 4.83, the entry in Table 1 of Bryant and Paulson (1976) and the
# multiplier behind the simultaneous intervals of the 1980 worked example.
cv_bryant_paulson(alpha_level = .05, df = 14, groups = 6, covariates = 1)

# With no covariates the reference distribution is the ordinary studentized
# range, which base R supplies directly. The critical value is then sqrt(2)
# times the Tukey HSD critical value.
cv_bryant_paulson(alpha_level = .05, df = 14, groups = 6, covariates = 0)$value
sqrt(2) * cv_tukey_hsd(alpha_level = .05, df = 14, groups = 6)$value

# The stepwise Duncan multiple-range significant range comes from
# procedure = "duncan". With covariates = 0 it reduces to Duncan's (1955)
# own significant studentized range, 3.37 for a stretch of 6 groups on 14
# error degrees of freedom.
cv_bryant_paulson(alpha_level = .05, df = 14, groups = 6, covariates = 0,
                  procedure = "duncan")
# One random covariate raises that range to 3.50, the entry in Table 2 of
# Bryant and Bruvold (1980). Being stepwise, the value is a running maximum
# of Bryant-Paulson quantiles, one inverted for every stretch from 2 to 6
# groups at that stretch's own protection level; the package tests check
# the full sequence against the paper's Section 4 example.
cv_bryant_paulson(alpha_level = .05, df = 14, groups = 6, covariates = 1,
                  procedure = "duncan")
```

cv_chisq *Provides the Critical Value(s) for a Chi Square Distribution*

Description

Provides the Critical Value(s) for a Chi Square Distribution

Usage

```
cv_chisq(
  alpha_level,
  df,
  alternative = "greater",
  alpha_lower,
  alpha_upper,
  ncp = 0,
  verbose = TRUE
)
```

Arguments

alpha_level	Type I error rate (i.e., the false positive rate).
df	The number of degrees of freedom (a positive number).
alternative	The type of alternative hypothesis of interest. The default, "greater", puts the whole of alpha_level in the upper tail, which is how the chi square distribution is used to test a model or an association (see Details).
alpha_lower	The error rate in the lower tail of the distribution.
alpha_upper	The error rate in the upper tail of the distribution.
ncp	The noncentral parameter (if zero, the default, it is the central chi square distribution).
verbose	Provides extra information about areas under the curve.

Details

Like the F distribution and unlike t and z, the chi square distribution is not symmetric and takes only non-negative values. Its common uses are one-sided in the upper tail: a test of association in a contingency table, a likelihood ratio test, and a test of model fit all reject for large values, because a poorly fitting model produces a large discrepancy, never a small one. That is why alternative defaults to "greater" here whereas it defaults to "not_equal" in cv_t. Maxwell, Delaney, and Kelley (2027) tabulate these upper-tail values in their Appendix Table A.9.

Both tails remain available for the situations that need them, most commonly an interval for a variance, which uses an upper and a lower chi square quantile. Set alternative = "not_equal", or give alpha_lower and alpha_upper directly. When a tail is given zero area its critical value is the boundary of the support, so lower_cv is 0 under the default.

A noncentral parameter can be supplied, which is what a power analysis for a test of model fit needs, though it would not be used for a standard null hypothesis significance test. See ci_nc_chisq for confidence limits on the noncentral parameter itself.

Value

Returns the critical value(s), based on the input specifications, in a output style (a data.frame with a row for the lower and the upper critical value, following the format used by cv_t).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (Appendix Table A.9 reports these critical values.)

See Also

cv_t, cv_f, ci_nc_chisq

Other critical values: cv_bonferroni_f(), cv_bryant_paulson(), cv_dunnett(), cv_f(), cv_scheffe(), cv_smm(), cv_t(), cv_tukey_hsd(), cv_z()

Examples

```
# The critical value for a test on 3 degrees of freedom at the .05 level.
cv_chisq(alpha_level = .05, df = 3)

# Simple output.
cv_chisq(alpha_level = .05, df = 3, verbose = FALSE)

# Both tails, as an interval for a variance would need.
cv_chisq(alpha_level = .05, df = 10, alternative = "not_equal")
```

---

cv_dunnett *Provides the Critical Value for Dunnett's Many-to-One Comparisons Procedure*

Description

Provides the Critical Value for Dunnett's Many-to-One Comparisons Procedure

Usage

```
cv_dunnett(
  alpha_level,
  df,
  n_comparisons,
  alternative = "not_equal",
  verbose = TRUE
)
```

Arguments

`alpha_level`	Type I error rate (i.e., the family-wise false-positive rate).
`df`	Error degrees of freedom (typically $N-k$, where k is the total number of groups including the control). May be `Inf`, in which case the known-variance (normal) limit is used.
`n_comparisons`	The number of treatment-versus-control comparisons (i.e., $k-1$ where k is the total number of groups).
`alternative`	The form of the alternative hypothesis: one of `"not_equal"` (two-sided; default), `"greater"` (treatments greater than control), or `"less"` (treatments less than control).
`verbose`	Provides extra information about areas under the curve.

Details

Dunnett's procedure controls the family-wise error rate for the special case of comparing each of $k-1$ treatments to a single control (the many-to-one comparisons setting), using a multivariate *t* reference with constant pairwise correlation $1/2$. That correlation is exact for a balanced design: each comparison is $(\bar{Y}_i - \bar{Y}_0)$, all sharing the one control mean $\bar{Y}_0$ of variance σ^2/n, while each comparison has variance $2\sigma^2/n$, so any two comparisons correlate $(\sigma^2/n)/(2\sigma^2/n) = 1/2$. Maxwell, Delaney, and Kelley (2027, Chapter 5) develop the many-to-one comparisons problem and tabulate these critical values.

How it is computed. The critical value is a quantile of a $(k-1)$-dimensional multivariate *t* distribution whose correlation matrix has a unit diagonal and off-diagonal entries of 1/2. Because that correlation is a single common value, the comparisons have the one-factor representation $Z_i = \sqrt{1/2}\,W + \sqrt{1/2}\,U_i$ with a shared factor W and independent U_i, all standard normal. Conditioning on W and on the common scale estimate $S = \sqrt{\chi^2_{df}/df}$ makes the comparisons independent, so the $(k-1)$-dimensional integral collapses to two nested one-dimensional integrals:

$$P\Big(\max_i T_i \le d\Big) = \int_0^\infty \int_{-\infty}^\infty \Big[\Phi\big((d\,s - \sqrt{1/2}\,w)/\sqrt{1/2}\big)\Big]^{k-1} \phi(w)\,dw\; f_S(s)\,ds$$

for the one-sided value (the two-sided value replaces the bracket with the probability that $|T_i| \le d$). This package evaluates those integrals with `integrate` and inverts with `uniroot`, so the returned value is deterministic and accurate to the solver tolerance, with no Monte Carlo simulation and no random seed. When $k-1=1$ the value is the ordinary one- or two-sided *t* critical value.

Value

Returns the critical value in a output style (a `data.frame` following the format used by `cv_t`).

Note

The constant-correlation assumption holds for balanced designs (equal n per group); for severely unbalanced designs use `glht` instead.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Dunnett, C. W. (1955). A multiple comparison procedure for comparing several treatments with a control. *Journal of the American Statistical Association, 50*(272), 1096–1121.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 5 on the multiple-comparisons problem; Dunnett's many-to-one comparisons procedure is developed there, and Appendix Tables A.6 and A.7 report its critical values for two- and one-tailed tests.)

See Also

`cv_t`, `cv_tukey_hsd`, `cv_smm`, `cv_scheffe`, `contrast_test`

Other critical values: `cv_bonferroni_f()`, `cv_bryant_paulson()`, `cv_chisq()`, `cv_f()`, `cv_scheffe()`, `cv_smm()`, `cv_t()`, `cv_tukey_hsd()`, `cv_z()`

Examples

```
# Following the many-to-one comparisons setting of Maxwell, Delaney, and
# Kelley (2027, Chapter 5): three treatment groups each compared to a single
# control (k = 4 groups, so 3 comparisons) with 36 error degrees of freedom.
# The two-sided critical value controls the family-wise error rate at .05.
cv_dunnett(alpha_level = .05, df = 36, n_comparisons = 3)

# When the treatments are expected only to exceed the control, a one-sided
# ("treatments greater than control") critical value is smaller.
cv_dunnett(alpha_level = .05, df = 36, n_comparisons = 3, alternative = "greater")
```

---

cv_f *Provides the Critical Value(s) for an* F *Distribution*

Description

Provides the Critical Value(s) for an *F* Distribution

Usage

```
cv_f(
  alpha_level,
  df_numerator,
  df_denominator,
  alternative = "greater",
  alpha_lower,
  alpha_upper,
  ncp = 0,
  verbose = TRUE
)
```

Arguments

alpha_level	Type I error rate (i.e., the false positive rate).
df_numerator	The numerator degrees of freedom (a positive number). In a model comparison this is the difference in the number of parameters between the two models.
df_denominator	The denominator (error) degrees of freedom (a positive number).
alternative	The type of alternative hypothesis of interest. The default, "greater", puts the whole of alpha_level in the upper tail, which is how the *F* distribution is used to test a model comparison (see Details).
alpha_lower	The error rate in the lower tail of the distribution.
alpha_upper	The error rate in the upper tail of the distribution.
ncp	The noncentral parameter (if zero, the default, it is the central *F* distribution).
verbose	Provides extra information about areas under the curve.

Details

Unlike the *t* and *z* distributions, the *F* distribution is not symmetric and takes only non-negative values, and the usual test of a model comparison is one-sided: a restricted model fits worse than a full model, so evidence against the restriction shows up as a large *F*, never a small one. That is why alternative defaults to "greater" here whereas it defaults to "not_equal" in cv_t. Maxwell, Delaney, and Kelley (2027) tabulate these upper-tail values in their Appendix Table A.2.

Both tails remain available for the situations that need them, such as an interval for a ratio of variances, either by setting alternative = "not_equal" or by giving alpha_lower and alpha_upper directly. When a tail is given zero area its critical value is the boundary of the support, so lower_cv is 0 under the default.

A noncentral parameter can be supplied, which is what a power analysis needs, though it would not be used for a standard null hypothesis significance test. See ci_nc_F for confidence limits on the noncentral parameter itself.

Value

Returns the critical value(s), based on the input specifications, in a output style (a data.frame with a row for the lower and the upper critical value, following the format used by cv_t).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3, where the *F* test of a model comparison is developed; Appendix Table A.2 reports these critical values.)

See Also

cv_t, cv_chisq, cv_bonferroni_f, cv_scheffe, ci_nc_F

Other critical values: cv_bonferroni_f(), cv_bryant_paulson(), cv_chisq(), cv_dunnett(), cv_scheffe(), cv_smm(), cv_t(), cv_tukey_hsd(), cv_z()

Examples

```
# The critical value for a model comparison with 3 numerator and 20
# denominator degrees of freedom, at the .05 level.
cv_f(alpha_level = .05, df_numerator = 3, df_denominator = 20)

# An omnibus test of four groups with 24 participants: a - 1 = 3 and
# N - a = 20 degrees of freedom; simple output.
cv_f(alpha_level = .05, df_numerator = 3, df_denominator = 20, verbose = FALSE)

# Both tails, as an interval for a ratio of variances would need.
cv_f(alpha_level = .05, df_numerator = 3, df_denominator = 20,
     alternative = "not_equal")
```

---

cv_scheffe *Provides the Critical Value for the Scheffé Procedure*

Description

Provides the Critical Value for the Scheffé Procedure

Usage

```
cv_scheffe(alpha_level, df_numerator, df_denominator, verbose = TRUE)
```

Arguments

alpha_level	Type I error rate (i.e., the family-wise false-positive rate).
df_numerator	The numerator degrees of freedom (typically the number of groups minus 1, $k - 1$, in a one-way ANOVA).
df_denominator	The denominator (error) degrees of freedom (typically $N - k$ in a one-way ANOVA).
verbose	Provides extra information about areas under the curve.

Details

The Scheffé critical value protects the family-wise error rate for the simultaneous test of *any* contrast (or family of contrasts) in a fixed-effects ANOVA, including data-driven contrasts selected after looking at the data. It is therefore the most conservative of the standard procedures.

The critical value, on the scale of a *t*-statistic, is

$$t_{\text{crit}}^{\text{Scheffe}} = \sqrt{(k-1)\, F_{1-\alpha,\, k-1,\, df_{\text{denominator}}}},$$

so that a contrast $\hat{\psi}$ with standard error $SE_{\hat{\psi}}$ is declared significant when $|\hat{\psi}/SE_{\hat{\psi}}| > t_{\text{crit}}^{\text{Scheffe}}$. The corresponding simultaneous confidence interval is $\hat{\psi} \pm t_{\text{crit}}^{\text{Scheffe}} \cdot SE_{\hat{\psi}}$.

Like the Studentized range distribution underlying Tukey HSD, the Scheffé reference is one-sided (the underlying *F* statistic is non-negative), so `alpha_level` is *not* split between two tails.

The Scheffé critical value is a function of a univariate *F* quantile, which base R supplies through `qf`, so unlike `cv_dunnett` and `cv_smm` this function needs no multivariate distribution machinery. Scheffé's procedure earns its simultaneous protection over the infinite family of all possible contrasts by projecting onto the overall *F* test rather than by integrating a multivariate *t* density; that is why a single univariate quantile suffices and the **mvtnorm** package is not required here. Maxwell, Delaney, and Kelley (2027, Chapter 5) develop the Scheffé method as the procedure for arbitrary contrasts within the multiple-comparisons problem.

Value

Returns the critical value in a output style (a `data.frame` with one row per critical value, following the format used by `cv_t` and `cv_tukey_hsd`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Scheffe, H. (1953). A method for judging all contrasts in the analysis of variance. *Biometrika, 40*, 87–104.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 5 on the multiple-comparisons problem, where the Scheffé method for arbitrary contrasts is developed.)

See Also

`cv_t`, `cv_tukey_hsd`, `contrast_test`

Other critical values: `cv_bonferroni_f()`, `cv_bryant_paulson()`, `cv_chisq()`, `cv_dunnett()`, `cv_f()`, `cv_smm()`, `cv_t()`, `cv_tukey_hsd()`, `cv_z()`

Examples

```
# Following the arbitrary-contrasts setting of Maxwell, Delaney, and Kelley
# (2027, Chapter 5): a one-way ANOVA with k = 4 groups and 36 error degrees
# of freedom (e.g., n = 10 per group). The Scheffé critical value protects
# the family-wise error rate over any contrast, including contrasts chosen
# after looking at the data.
cv_scheffe(alpha_level = .05, df_numerator = 3, df_denominator = 36)

# The price of that protection is a larger multiplier than the unadjusted
# t critical value at the same error degrees of freedom:
cv_t(alpha_level = .05, df = 36)
```

`cv_smm` *Provides the Critical Value of the Studentized Maximum Modulus Distribution*

Description

Provides the Critical Value of the Studentized Maximum Modulus Distribution

Usage

```
cv_smm(alpha_level, df, n_comparisons, verbose = TRUE)
```

Arguments

`alpha_level`	Type I error rate (i.e., the family-wise false-positive rate).
`df`	Error degrees of freedom (typically $N - k$ in a one-way ANOVA). May be `Inf`, in which case the known-variance (normal) limit is used.
`n_comparisons`	The number of simultaneous comparisons (m) for which a family-wise critical value is desired.
`verbose`	Provides extra information about areas under the curve.

Details

The Studentized maximum modulus (SMM) distribution is the distribution of

$$\max_{i=1,\ldots,m} |Z_i| \,/\, S,$$

where $Z_1, \ldots, Z_m$ are independent standard normal variates and S is independent of the Z_i and equal to $\sqrt{\chi^2_{df}/df}$.

What the SMM is used for. The SMM critical value is the multiplier that turns a set of m individual estimates into a family of simultaneous confidence intervals, or equivalently a family of tests, that jointly control the family-wise error rate at level `alpha_level`. Because the modulus is the largest of m standardized statistics in absolute value, requiring that maximum to clear the critical value

bounds the chance of any one of the m intervals failing to cover (or any one of the m tests producing a false positive). Maxwell, Delaney, and Kelley (2027, Chapter 5) describe this use in the context of the multiple-comparisons problem: when a researcher forms several means or contrasts and wants the stated coverage to hold across the whole set rather than one interval at a time, the SMM supplies the simultaneous critical value. A pair-by-pair construction, applied to m comparisons, is $\hat{\psi}_i \pm c_{\alpha;m,df}\, SE_{\hat{\psi}_i}$ with $c_{\alpha;m,df}$ the SMM critical value returned here. When $m = 1$ it reduces to the ordinary two-sided *t* critical value.

How it is computed. The m statistics are independent given the common scale estimate S, so the joint distribution factorizes after conditioning on S and the m-dimensional integral that defines the maximum modulus collapses to a single one-dimensional integral,

$$P\Big(\max_i |Z_i|/S \le c\Big) = \int_0^\infty \big[\, 2\,\Phi(c\,s) - 1 \,\big]^m \, f_S(s)\, ds,$$

where f_S is the density of $S = \sqrt{\chi^2_{df}/df}$. This package evaluates that integral with `integrate` and inverts it with `uniroot`, so the returned value is deterministic and accurate to the solver tolerance (there is no Monte Carlo simulation and no random seed). When `df = Inf` the scale is degenerate at 1 and the closed form $c = \Phi^{-1}\big((1 + (1-\alpha)^{1/m})/2\big)$ is returned; when $m = 1$ the value is exactly $t_{1-\alpha/2,\,df}$.

Value

Returns the critical value as a `data.frame`, following the format used by `cv_t`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Stoline, M. R., & Ury, H. K. (1979). Tables of the Studentized maximum modulus distribution and an application to multiple comparisons among means. *Technometrics, 21*(1), 87–93.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 5 on the multiple-comparisons problem, where simultaneous confidence intervals for several means or contrasts are developed; Appendix Table A.5 reports SMM critical values.)

See Also

`cv_t`, `cv_tukey_hsd`, `cv_dunnett`, `cv_scheffe`

Other critical values: `cv_bonferroni_f()`, `cv_bryant_paulson()`, `cv_chisq()`, `cv_dunnett()`, `cv_f()`, `cv_scheffe()`, `cv_t()`, `cv_tukey_hsd()`, `cv_z()`

Examples

```
# Following the simultaneous-intervals setting of Maxwell, Delaney, and
# Kelley (2027, Chapter 5): a researcher forms m = 5 comparisons and wants
# all five confidence intervals to hold simultaneously at the .05 level,
# with 36 error degrees of freedom.
```

```
cv_smm(alpha_level = .05, df = 36, n_comparisons = 5)

# When m = 1, the SMM critical value reduces to the two-sided
# t critical value:
cv_smm(alpha_level = .05, df = 36, n_comparisons = 1)$value
cv_t(alpha_level = .05, df = 36)$value[2]   # upper_cv from cv_t
```

---

cv_t *Provides the Critical Value(s) for a* t*-distribution*

Description

Provides the Critical Value(s) for a *t*-distribution

Usage

```
cv_t(
  alpha_level,
  df,
  alternative = "not_equal",
  alpha_lower,
  alpha_upper,
  ncp = 0,
  verbose = TRUE
)
```

Arguments

alpha_level	Type I error rate (i.e., the false positive rate).
df	The number of degrees of freedom (a positive number)
alternative	The type of alternative hypothesis of interest.
alpha_lower	The error rate on the lower (negative) side of the distribution.
alpha_upper	The error rate on the upper (positive) side of the distribution.
ncp	The noncentral parameter (if zero, the default, it is the central *t*-distribution).
verbose	Provides extra information about areas under the curve.

Details

Though a noncentral parameter can be included, that would not be done for a standard null hypothesis significance test.

Value

Returns the critical value(s), based on the input specifications, in a output style.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

Other critical values: cv_bonferroni_f(), cv_bryant_paulson(), cv_chisq(), cv_dunnett(), cv_f(), cv_scheffe(), cv_smm(), cv_tukey_hsd(), cv_z()

Examples

```
# A basic call for finding critical values with equal area in the two tails.
cv_t(alpha_level = .05, df = 13)

# A basic call for a single-sided confidence interval (for "a greater than" alternative hypothesis)
cv_t(alpha_level = .05, df = 13, alternative = "greater")

# A single-sided confidence interval (for "a greater than" alternative hypothesis); simple output.
cv_t(alpha_lower = 0, alpha_upper = .05, df = 13, verbose = FALSE)

# For a nonsymmetric 95% confidence interval.
cv_t(alpha_lower = .01, alpha_upper = .04, df = 13)
```

---

cv_tukey_hsd *Provides the Critical Value for the Tukey Honestly Significant Difference (HSD) Test*

Description

Provides the Critical Value for the Tukey Honestly Significant Difference (HSD) Test

Usage

```
cv_tukey_hsd(alpha_level, df, groups, verbose = TRUE)
```

Arguments

alpha_level	Type I error rate (i.e., the false positive rate). For the Tukey HSD test, the full alpha_level applies to the upper tail of the Studentized range distribution (see Details).
df	The error degrees of freedom from the ANOVA (a positive number; typically N - groups).

groups	The number of groups whose means are being compared (an integer of at least 2).
verbose	Provides extra information about areas under the curve.

Details

The Tukey HSD test compares all pairs of group means using the Studentized range distribution (qtukey). Because the Studentized range is the absolute difference between the largest and smallest sample means (scaled by a standard error), it is non-negative and the associated distribution has support on $[0, \infty)$. As a consequence, the Type I error rate alpha_level is *not* split between two tails in the way it is for the (symmetric) *t*- and *z*-distributions in cv_t and cv_z; rather, the full alpha_level applies to the upper tail.

The reported critical value is on the scale used for pairwise comparisons of group means, i.e., $q_{1-\alpha,k,df}/\sqrt{2}$, where k is the number of groups. A pair of means is declared significantly different when the absolute standardized difference between them exceeds this critical value.

The Tukey HSD critical value is a quantile of the Studentized range distribution, which base R supplies through qtukey, so unlike cv_dunnett and cv_smm this function needs no multivariate distribution machinery and does not require the **mvtnorm** package. Maxwell, Delaney, and Kelley (2027, Chapter 5) develop the Tukey method as the procedure for all-pairwise comparisons within the multiple-comparisons problem.

Value

Returns the critical value in a output style (a data.frame with one row per critical value, following the format used by cv_t and cv_z).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Tukey, J. W. (1953). *The problem of multiple comparisons*. Unpublished manuscript, Princeton University.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 5 on the multiple-comparisons problem, where the Tukey method for all-pairwise comparisons is developed.)

See Also

cv_t, cv_z, TukeyHSD, qtukey

Other critical values: cv_bonferroni_f(), cv_bryant_paulson(), cv_chisq(), cv_dunnett(), cv_f(), cv_scheffe(), cv_smm(), cv_t(), cv_z()

Examples

```
# Following the all-pairwise comparisons setting of Maxwell, Delaney, and
# Kelley (2027, Chapter 5): every pair of k = 3 group means is compared,
# with 27 error degrees of freedom, holding the family-wise error rate at
# .05.
cv_tukey_hsd(alpha_level = .05, df = 27, groups = 3)

# Using DMAR's test_market data (6 marketing panels, N = 24).
fit <- aov(brand_movement ~ panel, data = test_market)
cv_tukey_hsd(
  alpha_level  = .05,
  df     = df.residual(fit),
  groups = nlevels(test_market$panel)
)

# A more stringent alpha with a simple (non-verbose) result.
cv_tukey_hsd(alpha_level = .01, df = 27, groups = 3, verbose = FALSE)
```

---

cv_z *Provides the Critical Value(s) for the Standard Normal Distribution (the z-distribution, With Mean 0 and Variance 1)*

Description

Provides the Critical Value(s) for the Standard Normal Distribution (the z-distribution, With Mean 0 and Variance 1)

Usage

```
cv_z(
  alpha_level,
  alternative = "not_equal",
  alpha_lower,
  alpha_upper,
  verbose = TRUE
)
```

Arguments

alpha_level	Type I error rate (i.e., the false positive rate).
alternative	The type of alternative hypothesis of interest.
alpha_lower	The error rate on the lower (negative) side of the distribution.
alpha_upper	The error rate on the upper (positive) side of the distribution.
verbose	Provides extra information about areas under the curve.

Value

Returns the critical value(s), based on the input specifications, in a output style.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

Other critical values: cv_bonferroni_f(), cv_bryant_paulson(), cv_chisq(), cv_dunnett(), cv_f(), cv_scheffe(), cv_smm(), cv_t(), cv_tukey_hsd()

Examples

```
# A basic call for finding critical values with equal area in the two tails.
cv_z(alpha_level = .05)

# A basic call for a single-sided confidence interval (for "a greater than" alternative hypothesis)
cv_z(alpha_level = .05, alternative = "greater")

# A single-sided confidence interval (for "a greater than" alternative hypothesis); simple output.
cv_z(alpha_lower = 0, alpha_upper = .05, verbose = FALSE)

# For a nonsymmetric 95% confidence interval.
cv_z(alpha_lower = .01, alpha_upper = .04)
```

---

depression_bdi *Depression Treatment Study With a Pretest Covariate*

Description

The hypothetical three-group depression study that runs through the analysis of covariance development of Maxwell, Delaney, and Kelley (2027, Chapter 9, Table 9.7). Thirty depressive individuals are randomly assigned, ten per group, to a selective serotonin reuptake inhibitor (SSRI), a placebo, or a wait list control. The Beck Depression Inventory (BDI) is administered before the study begins and again at its end, giving a pretest that serves as the covariate and a posttest that serves as the outcome.

Usage

```
depression_bdi
```

Format

A `data.frame` with 30 rows and 3 columns:

condition Factor with levels `ssri`, `placebo`, and `wait_list`: the randomly assigned treatment.

bdi_pre Beck Depression Inventory score before the study.

bdi_post Beck Depression Inventory score at the end of the study.

Details

This is the worked example behind the ANCOVA contrast pages: the pretest group means are 17, 17.7, and 17.4; the within-groups sum of squares of the pretest is 752.5; the ANCOVA error variance is about 29; and the covariate-adjusted posttest means are approximately 7.5, 12, and 14 for the SSRI, placebo, and wait list groups. The examples of `ci_c_ancova` and `ci_sc_ancova` quote those summary values, and a test recomputes each of them from these data so the printed numbers cannot drift.

The data are hypothetical, constructed for the book; higher BDI scores mean more severe depressive symptoms, so a treatment that works pulls the posttest down. The same numeric values ship in the book's data companion, the **AMCP** package, as `chapter_9_table_7`; DMAR carries them directly so its ANCOVA examples and tests need no package beyond this one.

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9 on ANCOVA.)

Examples

```
data(depression_bdi)

# The fingerprints the ANCOVA contrast pages quote.
tapply(depression_bdi$bdi_pre, depression_bdi$condition, mean)
fit <- lm(bdi_post ~ bdi_pre + condition, data = depression_bdi)
anova(fit)

# The covariate-adjusted group means at the pretest grand mean.
ancova(depression_bdi, outcome = "bdi_post", treatment = "condition",
       covariates = "bdi_pre")
```

---

descriptives *Descriptive Statistics for One or More Variables*

Description

Computes a compact set of descriptive statistics useful for data screening and psychometric work (including skewness and excess kurtosis), with an optional correlation matrix for the numeric variables.

Usage

```
descriptives(x, correlations = FALSE, listwise = FALSE)
```

Arguments

x	A data.frame, tibble, or matrix containing the variables to summarize. A matrix is coerced to a data.frame.
correlations	Logical. If TRUE, also return a correlation matrix for the numeric variables (see Details).
listwise	Logical. If TRUE, apply listwise deletion (drop every row containing any NA) to x *before* computing any statistics. If FALSE (the default), each variable is summarized using its own available (non-missing) observations.

Details

Skewness and kurtosis. The reported values use the bias-corrected formulas commonly referred to as SAS/SPSS Type 2:

$$\text{skewness} = \frac{n}{(n-1)(n-2)} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s} \right)^3,$$

$$\text{kurtosis} = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s} \right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)},$$

where s is the (divisor-$n-1$) sample standard deviation. Kurtosis is reported as excess kurtosis, so a normal distribution has an expected value of 0. As a rough guide to deciding whether normal-theory inference (e.g., maximum likelihood in factor analysis or SEM) is defensible, values of $|\text{skewness}| > 2$ or $|\text{kurtosis}| > 7$ are frequently flagged as problematic.

Variable-type handling. Each column is classified as "numeric", "integer", "logical", "factor", "character", or whatever its first class is otherwise. Columns of type "numeric", "integer", and "logical" receive full distributional summaries (logicals are coerced so that mean is the proportion of TRUEs). All other columns report only n, n_missing, and prop_missing; their remaining columns are NA.

Correlations. The correlation matrix, when requested, uses Pearson correlations with use = "pairwise.complete.obs" so that each pairwise correlation uses the maximum information available. Only numeric-type variables are included.

Value

A list with two elements, returned in a stable shape regardless of the correlations argument:

descriptives A data.frame with one row per input variable and columns variable, type, n, n_missing, prop_missing, mean, median, sd, min, max, q25, q75, skewness, and kurtosis (excess kurtosis). For non-numeric variables, the numeric summary columns are NA.

correlations A $p \times p$ correlation matrix among the numeric variables, or NULL if correlations = FALSE or fewer than two numeric variables are available. This is returned as a plain matrix (not a table), because its natural structure is a symmetric two-dimensional array; that format reads well and plugs directly into downstream tools such as cov2cor or factor analysis and SEM software.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

cor, quantile

Other descriptive statistics: kurtosis(), skewness()

Examples

```
# Four cognitive tests from the Holzinger and Swineford (1939) study
# (301 children in two schools).
hs_tests <- holzinger_swineford[, c("t1_visual_perception", "t2_cubes",
                                    "t4_lozenges",
                                    "t6_paragraph_comprehension")]
descriptives(hs_tests)

# Include a correlation matrix (useful during scale development).
descriptives(hs_tests, correlations = TRUE)

# Mixed-type data: numeric summaries for the test scores, and
# type = "factor" (with NA numeric columns) for school.
descriptives(holzinger_swineford[, c("school", "t1_visual_perception",
                                     "t2_cubes")])

# Data with missing values: the revised paper form board and flags tests
# were administered only in the Grant-White school, so 156 of the 301
# children have no score. Per-variable N and missingness are reported.
hs_partial <- holzinger_swineford[, c("t1_visual_perception",
                                      "t25_paper_form_board_r",
                                      "t26_flags")]
descriptives(hs_partial)

# The same data with listwise deletion applied first.
descriptives(hs_partial, listwise = TRUE)
```

---

design_consequences *Consequences of a Design: Power, Sign and Magnitude Errors, and Expected Precision*

Description

Evaluates what a design of a given precision will actually deliver, under both of the package's lenses at once. The *significance lens*: the power of the two-sided test, the type_s_error (the probability that a statistically significant estimate has the wrong sign), and the exaggeration_ratio (Type M: the average factor by which significant estimates overstate the true effect), following the design analysis of Gelman and Carlin (2014). The *precision lens*, in the accuracy in parameter estimation (AIPE) tradition: the expected half-width and full width of the conf_level confidence interval the design will produce, the spread of that realized width, and, when a target width w is supplied, pct_ci_less_w, the probability that the realized interval is no wider than the target, the same quantities the ss_aipe_*_sensitivity() family estimates by Monte Carlo, here in closed form.

Usage

```
design_consequences(
  true_effect = NULL,
  se = NULL,
  sd = NULL,
  n_1 = NULL,
  n_2 = NULL,
  alpha_level = 0.05,
  df = NULL,
  conf_level = 0.95,
  w = NULL
)
```

Arguments

`true_effect`	The assumed true (population) effect, on the scale of the estimate (a mean difference, a regression coefficient, a standardized mean difference). May be negative. May be `NULL`, in which case the significance-lens rows are returned as `NA` and only the precision lens (which does not involve the true effect) is informative.
`se`	The standard error of the estimate the design will produce, on the same scale as `true_effect`. Supply `se` directly, or supply `sd` with `n_1` (and `n_2`) and let the function derive it.
`sd`, `n_1`, `n_2`	An alternative to `se` for the two most common cases: with `sd` and `n_1` only, the one-sample (or paired-difference) design `se = sd / sqrt(n_1)` with `df = n_1 - 1`; with `n_2` as well, the two-group design `se = sd * sqrt(1/n_1 + 1/n_2)` with `df = n_1 + n_2 - 2` (`sd` is the common within-group standard deviation). A `df` supplied explicitly overrides the derived one.
`alpha_level`	Two-sided Type I error rate of the significance test. Defaults to 0.05.
`df`	Degrees of freedom of the reference *t* distribution. Defaults to `Inf` (the normal case) unless derived from `n_1` / `n_2`.
`conf_level`	Confidence level of the interval evaluated by the precision lens. Defaults to 0.95.
`w`	Optional target full width for the confidence interval; when supplied, `pct_ci_less_w` reports the probability that the realized interval is no wider than `w`.

Details

Together they answer the two questions a chosen design should be interrogated with before data collection: *if I run this study and filter it through a significance test, what will the published record look like?* and *how precisely will I estimate the effect regardless of significance?* An underpowered design fails both: its significant estimates are exaggerated and possibly sign-reversed, and its confidence intervals are too wide to be informative.

Significance lens. Writing $\lambda = \theta/\mathrm{se}$ and c for the two-sided critical value, the power and the Type S error follow from the two tails of the distribution of the test statistic: the noncentral *t* with noncentrality λ when `df` is finite (the exact distribution of the *t* statistic when the standard error is estimated from the data, the same sampling model the precision lens uses), and the normal when `df = Inf`. The exaggeration ratio is the expected absolute estimate conditional on significance over

the absolute true effect, computed exactly: from truncated normal moments when `df = Inf`, and otherwise by integrating those moments over the chi distribution of the estimated standard error, so no simulation error enters. Gelman and Carlin's `retrodesign()` instead evaluates a central t shifted by λ (and simulates the exaggeration ratio under that model), an approximation that treats the standard error as known; the two agree as `df` grows and coincide at `df = Inf`, but at small `df` they differ: in the underpowered regime the design analysis is aimed at (power below about 0.7), the known-se approximation understates power and overstates the Type S and Type M errors, so their published finite-`df` values differ from the exact ones reported here. When `true_effect = 0` the power equals `alpha_level`, the Type S error is 0.5, and the exaggeration ratio is undefined (`NA`).

Precision lens. The realized interval half-width is $t_{1-\alpha^*/2,\,df} \cdot \widehat{\mathrm{se}}$ with $\alpha^* = 1 - \texttt{conf\_level}$, and $\widehat{\mathrm{se}} = \mathrm{se}\sqrt{W/df}$ with $W \sim \chi^2_{df}$, so the width's mean, median, and standard deviation have closed chi-distribution forms and

$$P(\text{width} \le w) \;=\; P\!\left(W \le df\left[\frac{w}{2\,t\,\mathrm{se}}\right]^2\right).$$

These are the population versions of the `mean_ci_width`, `median_ci_width`, `sd_ci_width`, and `pct_ci_less_w` terms that the `ss_aipe_*_sensitivity()` functions estimate by Monte Carlo. With `df = Inf` the standard error is treated as known, the width is deterministic, and `pct_ci_less_w` is a step: 1 when the fixed width is at most `w` and 0 otherwise.

The function complements the planners rather than replacing them: `ss_power_*()` chooses a sample size for detection, `ss_aipe_*()` chooses one for precision, and `design_consequences()` interrogates whatever design came out (or the design a completed study used). The two lenses are the power and accuracy in parameter estimation approaches to sample size planning reviewed by Maxwell, Kelley, and Rausch (2008).

Value

A `data.frame` (class `dmar_tbl`) with the significance-lens rows (`power`, `type_s_error`, `exaggeration_ratio`), the precision-lens rows (`expected_half_width`, `mean_ci_width`, `median_ci_width`, `sd_ci_width`, `pct_ci_less_w`, `target_width`; the last two are `NA` when no `w` is supplied), and the design rows (`true_effect`, `se`, `df`, `alpha_level`). The confidence level is recorded in the `"conf_level"` attribute. The schema is constant: rows that do not apply are `NA`, never dropped.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Gelman, A., & Carlin, J. (2014). Beyond power calculations: Assessing Type S (sign) and Type M (magnitude) errors. *Perspectives on Psychological Science, 9*(6), 641–651. doi:10.1177/1745691614551642 (Their accompanying `retrodesign()` function evaluates a location-shifted central *t*, the known-se approximation, and obtains the exaggeration ratio by simulation; the finite-df case here uses the exact noncentral *t* and exact moments instead, so the two differ at small df. See Details.)

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Kelley, K., Maxwell, S. E., & Rausch, J. R. (2003). Obtaining power or obtaining precision: Delineating methods of sample size planning. *Evaluation and the Health Professions, 26*(3), 258–287. doi:10.1177/0163278703255242

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

See Also

ss_power_smd and ss_aipe_smd for choosing the sample size by detection or by precision before this function interrogates the choice; expected_smd for the unconditional small-sample bias of the standardized mean difference, a different bias than the significance-filter exaggeration here.

Other design utilities: design_effect(), effects_coding(), helmert_coding(), is_orthogonal_set(), orthogonal_polynomial()

Examples

```
# ---- Both lenses on one underpowered design --------------------------
# True effect 0.1 measured with standard error 0.3 (say, d = .1 with
# about 22 per group, 45 in total): power is 6 percent, a significant
# result has a 17 percent chance of the wrong sign and overstates the
# truth seven-fold, and the 95 percent CI is about 1.2 wide, twelve
# times the effect. Bad for detection, bad for precision.
design_consequences(true_effect = 0.1, se = 0.3)

# ---- The same effect, precisely measured -----------------------------
design_consequences(true_effect = 0.1, se = 0.03)

# ---- From a planned two-group design (sd and per-group n) ------------
# d-type effect 0.4, common sd 1, 60 per group; finite df flows through
# both lenses.
design_consequences(true_effect = 0.4, sd = 1, n_1 = 60, n_2 = 60)

# ---- Will the interval beat a target width? --------------------------
# Same design, asking for the probability the realized 95 percent CI is
# no wider than 0.7 (the closed-form pct_ci_less_w that the
# ss_aipe_smd_sensitivity() simulation estimates).
design_consequences(true_effect = 0.4, sd = 1, n_1 = 60, n_2 = 60,
                    w = 0.7)

# ---- Precision lens alone (no effect assumption needed) --------------
design_consequences(true_effect = NULL, sd = 1, n_1 = 60, n_2 = 60,
                    w = 0.7)

# ---- After an AIPE plan: check the detection side --------------------
# Plan for a full width of 0.5 on the SMD at delta = .4, then ask what
# that design does under the significance filter.
n_plan <- ss_aipe_smd(delta = 0.4, width = 0.5)$value[1]
design_consequences(true_effect = 0.4, sd = 1,
```

```
n_1 = n_plan, n_2 = n_plan, w = 0.5)
```

`design_effect` *Kish's Design Effect (DEFF), DEFT, and the Effective Sample Size*

Description

Computes the design effect (DEFF) and its square root (DEFT) for a clustered or multistage sample, given a vector of per-cluster sample sizes and a value of the intraclass correlation. Returns Kish's (1965) classic formula together with the effective sample size and a description of the clustering, including the number of empty clusters (no observations) and the number of singleton clusters (one observation), which carry different amounts of within-cluster information in a mixed-effects context.

Usage

```
design_effect(cluster_sizes, icc)
```

Arguments

`cluster_sizes`	Numeric vector of per-cluster sample sizes. Each element is the number of observations in one cluster (so the length of the vector is the number of clusters). Zero values are allowed and counted as empty clusters; they do not affect the computation of DEFF.
`icc`	Intraclass correlation coefficient, in $[0, 1)$. Use the population value when planning prospectively, or the sample estimate (e.g., from `icc` or `icc_lmer`) when describing an observed clustered sample.

Details

Definition (Kish, 1965). For a clustered sample with per-cluster sizes $m_1, m_2, \ldots, m_K$ and intraclass correlation ρ, the design effect on the variance of the mean is

$$\mathrm{DEFF} \;=\; 1 + (\bar{m}^{*} - 1)\rho,$$

where $\bar{m}^{*} = \sum_k m_k^2 / \sum_k m_k$ is the design-weighted average cluster size (Kish, 1965, eq. 5.4; sometimes called the "Kish weighted average" or "effective cluster size"). For equal cluster sizes $m_k = m$, this reduces to the classroom form $1 + (m - 1)\rho$. The DEFT is the square root of DEFF and is the inflation factor on the *standard error* of the mean (whereas DEFF inflates the variance).

Effective sample size. The number of observations from a simple random sample that would yield the same standard error as the clustered sample is

$$N_{\mathrm{eff}} \;=\; N/\mathrm{DEFF} \;=\; N/\mathrm{DEFT}^2,$$

where $N = \sum_k m_k$ is the total observations.

Why empty and singleton clusters are reported separately. In mixed-effects / multilevel modeling, clusters with zero observations carry no information (they should be dropped before fitting),

and clusters with one observation contribute to N and to the fixed-effect estimate but contribute nothing to the estimation of the random-effect variance or to the within-cluster residual. Hox et al. (2017) note that singleton-heavy designs have a design effect close to 1 even at moderate ρ because the weighted cluster size is small. The output reports the counts of empty and singleton clusters so the user can see at a glance how much of the nominal sample size carries clustering information.

If a fitted `lmerMod` object is available, the typical workflow is to extract `icc` via `icc_lmer` and the per-cluster sample sizes via `table(cluster_id)`, then pass both to `design_effect()`; see the second example.

Value

A `data.frame` with rows for the design effect, its square root, the effective sample size, the total observation and cluster counts (including separate counts of empty and singleton clusters), the mean cluster size, Kish's design-weighted mean cluster size, and the input `icc`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Hox, J. J., Moerbeek, M., & van de Schoot, R. (2017). *Multilevel analysis: Techniques and applications* (3rd ed.). Routledge.

Kish, L. (1965). *Survey sampling*. Wiley.

Kish, L. (1992). Weighting for unequal Pi. *Journal of Official Statistics, 8*(2), 183–200.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapters 15 and 16 on mixed-effects models and nested designs.)

Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel analysis: An introduction to basic and advanced multilevel modeling* (2nd ed.). Sage.

See Also

`icc`, `icc_lmer`, `var_icc`, `ss_aipe_icc`

Other design utilities: `design_consequences()`, `effects_coding()`, `helmert_coding()`, `is_orthogonal_set()`, `orthogonal_polynomial()`

Examples

```
# 1. Balanced design: K = 30 clusters of size 20, ICC = 0.10.
design_effect(cluster_sizes = rep(20, 30), icc = 0.10)
# DEFF = 1 + (20 - 1) * 0.10 = 2.9; DEFT = 1.7; effective N = 600 / 2.9 = 207.

# 2. Unbalanced design with some empty and some singleton clusters.
# Suppose K = 25 schools with attendance ranging from 0 (closed)
# through 1 (single student showed up) to 30 (full class).
sizes <- c(0, 0, 1, 1, 1, 2, 3, 5, 8, 10, 12, 15, 18, 20, 22,
           22, 25, 26, 28, 28, 30, 30, 30, 30, 30)
```

```
design_effect(cluster_sizes = sizes, icc = 0.10)

# 3. From a cluster-id vector: tabulate, then call design_effect().
cluster_id <- rep(1:8, times = c(20, 15, 22, 1, 0, 30, 18, 25))
design_effect(cluster_sizes = as.numeric(table(factor(cluster_id, levels = 1:8))),
     icc = 0.15)
```

`diagnosis_agreement` *Cohen's (1968) Psychiatric Diagnosis Agreement Table*

Description

The illustrative agreement matrix from Cohen's (1968) weighted kappa paper, Table 1: two judges independently assign $N = 200$ cases to three diagnostic categories (personality disorder, neurosis, psychosis). The data set reproduces the printed table cell for cell and in its original layout, Judge B indexing the rows and Judge A the columns, one row per cell of the 3 x 3 matrix. Each cell carries the three quantities Cohen prints: the ratio-scaled disagreement weight, the chance-expected proportion (his parenthetical values), and the observed proportion; the raw frequency is the observed proportion times *N*.

Usage

```
diagnosis_agreement
```

Format

A data frame with 9 observations (one per cell of the 3 x 3 agreement matrix) on 6 variables.

`judge_b` Factor: Judge B's diagnostic category (the table's rows), with levels `Personality disorder`, `Neurosis`, `Psychosis`.

`judge_a` Factor: Judge A's diagnostic category (the table's columns), same levels.

`frequency` Number of the 200 cases jointly assigned to the cell.

`disagreement_weight` Cohen's ratio-scaled disagreement weight v_{ij} for the cell: 0 on the agreement diagonal, 1 for a personality disorder-neurosis confusion, 3 for personality disorder-psychosis, and 6 for neurosis-psychosis, the confusion the illustration treats as gravest.

`observed_proportion` Observed proportion of cases in the cell, `frequency / 200`.

`expected_proportion` Chance-expected proportion of cases in the cell, the product of the cell's row (Judge B) and column (Judge A) marginal proportions; the parenthetical values in Cohen's Table 1.

Details

Cohen built this table to make a point that is easy to miss: weighted kappa is fully chance corrected, and it can be *smaller* than unweighted kappa on the same data. Here the judges disagree far less than chance expectation in the mildly weighted personality disorder-neurosis cells but at about the chance level in the heavily weighted neurosis-psychosis cells, so $\kappa = .492$ while $\kappa_W = .348$: they disagree least where it matters least. Interchanging the 6 and 1 weights reverses the conclusion ($\kappa_W = .574$).

The reconstruction was verified against every quantity Cohen computes from the table: the marginals (.50/.30/.20 for Judge B, .60/.30/.10 for Judge A), the chance-expected cell proportions, the weighted disagreement sums $q'_o = .90$ and $q'_c = 1.38$, $\kappa = .492$, $\kappa_W = .348$, and his Formula 10 and 13 standard errors (.0901 and .0916). The cohen_kappa help page replicates the full set of analyses, and the weighted kappa vignette works the illustration end to end, including the orientation of the printed weight display in the paper's asymmetric-weight validity reinterpretation.

Author(s)

Ken Kelley <kkelley@nd.edu>

Source

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin, 70*(4), 213–220 (Table 1, p. 214).

References

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin, 70*(4), 213–220. doi:10.1037/h0026256

See Also

cohen_kappa, which analyzes this table in its examples.

Other reliability: cohen_kappa(), fleiss_kappa(), icc(), reliability(), reliability_H(), reliability_alpha(), reliability_kr20(), reliability_omega(), reliability_omega_categorical()

Examples

```
data(diagnosis_agreement)

# Rebuild Cohen's Table 1 layout (Judge B in rows, Judge A in columns).
xtabs(frequency ~ judge_b + judge_a, data = diagnosis_agreement)

# Unweighted and weighted kappa, reproducing kappa = .492 and
# kappa_w = .348.
tab <- xtabs(frequency ~ judge_b + judge_a, data = diagnosis_agreement)
v   <- xtabs(disagreement_weight ~ judge_b + judge_a,
             data = diagnosis_agreement)
cohen_kappa(table = tab)
cohen_kappa(table = tab, weights = unclass(v),
            weight_scaling = "disagreement")
```

```
# Cohen's Formula 8 directly from the per-cell quantities.
with(diagnosis_agreement,
     1 - sum(disagreement_weight * observed_proportion) /
         sum(disagreement_weight * expected_proportion))
```

---

dmacs *The dMACS Effect Size of Measurement Noninvariance*

Description

Quantifies how much a violation of measurement invariance actually matters for an item, rather than only whether it is statistically detectable. A likelihood ratio or score test can flag a loading or intercept difference that is too small to change anyone's score in a meaningful way, and with a large sample size it usually will. The dMACS index of Nye and Drasgow (2011) answers the size question directly: it is the expected difference between the reference group's and the focal group's measurement equations for that item, averaged over the focal group's latent distribution and standardized by the pooled item standard deviation, so it reads on the familiar standardized mean difference scale. Input is either a fitted multiple group **lavaan** model (requires **lavaan**) or the loadings, intercepts, and pooled standard deviations a paper reports.

Usage

```
dmacs(
  fit = NULL,
  reference = NULL,
  focal = NULL,
  lambda_reference = NULL,
  lambda_focal = NULL,
  nu_reference = NULL,
  nu_focal = NULL,
  mean_focal = 0,
  sd_focal = 1,
  sd_pooled = NULL,
  item_names = NULL
)
```

Arguments

`fit`	A fitted multiple group **lavaan** object carrying a mean structure, with the two groups' loadings and intercepts on a common metric (see Details). Supply either `fit` or the parameter vectors, never both.
`reference, focal`	Which groups play the reference and focal roles, each given as a single group label or a single group index in `lavInspect(fit, "group.label")`. When both are `NULL` (default) and the fit has exactly two groups, the first is the reference and the second the focal; with more than two groups they must be named.

`lambda_reference`, `lambda_focal`
: Numeric vectors of unstandardized loadings, one per item, in the reference and focal groups. Any finite values are admissible.

`nu_reference`, `nu_focal`
: Numeric vectors of unstandardized intercepts, one per item, in the reference and focal groups, in the same item order as the loadings. Any finite values are admissible.

`mean_focal`
: Focal group latent mean μ_F on the common metric, a single finite number (default `0`, the usual identification in which the reference group's latent mean is fixed at zero). Used only on the parameter vector path; with a `fit` it is read from the fit.

`sd_focal`
: Focal group latent standard deviation σ_F on the common metric, a single positive number (default `1`). Used only on the parameter vector path; with a `fit` it is read from the fit.

`sd_pooled`
: Pooled observed standard deviation of each item across the two groups, either one positive number applied to every item or one per item. Required on the parameter vector path; with a `fit` it is computed from the group sample sizes and observed item variances.

`item_names`
: Optional character vector of item labels, one per item. Defaults to the names carried by the parameter vectors, to the indicator names in the fit, or to `item_1`, `item_2`, and so on.

Details

For item i, let ν_R and λ_R be the reference group's intercept and loading, let ν_F and λ_F be the focal group's, and let the focal group's latent variable be $\eta \sim N(\mu_F, \sigma_F^2)$ with density f_F. Each group's measurement equation gives an expected item score at every value of η, and dMACS is the root mean squared vertical distance between those two lines over the focal group's latent distribution, divided by the pooled item standard deviation:

$$d_{MACS,i} = \frac{1}{SD_i}\sqrt{\int \left[(\nu_R + \lambda_R\eta) - (\nu_F + \lambda_F\eta)\right]^2 f_F(\eta)\, d\eta}.$$

Writing $a = \nu_R - \nu_F$ for the intercept difference and $b = \lambda_R - \lambda_F$ for the loading difference, the integrand is $(a + b\eta)^2$ and the integral is the second moment of a linear function of a normal variate, so it has the closed form $a^2 + 2ab\mu_F + b^2(\sigma_F^2 + \mu_F^2)$. No numerical integration is performed. The standardizer is the pooled observed standard deviation of the item,

$$SD_i = \sqrt{\frac{(n_R - 1)s_R^2 + (n_F - 1)s_F^2}{n_R + n_F - 2}},$$

the same pooling used by the standardized mean difference, which puts dMACS on a scale a reader of `smd` already understands.

dMACS is a root mean square and so is nonnegative by construction; it carries no sign and is not given one here. The direction of the violation is read from the returned components: $\nu_R - \nu_F$ says which group is scored higher at the mean of the latent variable, and $\lambda_R - \lambda_F$ says in which group

the item discriminates more sharply. When only the intercepts differ, the integral collapses to a^2 and dMACS reduces to $|a|/SD_i$, a plain standardized intercept difference.

The index is interpretable only when the two groups' loadings and intercepts are expressed on a common metric. In practice that means a partial invariance model in which a set of anchor items is constrained equal across groups while the suspect items are freed, and the focal group's latent mean and variance are freely estimated. A configural model, which sets each group's latent scale separately, does not put the groups on a common metric, and dMACS computed from one is not meaningful. The usual workflow is therefore `measurement_invariance` to locate where the ladder breaks, a partial invariance refit that frees the offending parameters, and then `dmacs()` on that refit to judge whether the violation is large enough to matter.

On the fit path each group's item variance is computed from the data in the fit with the usual $n - 1$ divisor, using the number of nonmissing observations for that item in that group. When the model was fitted from sample moments rather than raw data, the sample covariances in the fit are used instead, rescaled to the $n - 1$ divisor when the fit's likelihood option calls for it.

Value

A wide `data.frame` (class `dmar_tbl`) with one row per item and columns

`item` Item label.

`lambda_reference` Reference group unstandardized loading.

`lambda_focal` Focal group unstandardized loading.

`nu_reference` Reference group unstandardized intercept.

`nu_focal` Focal group unstandardized intercept.

`sd_pooled` Pooled observed standard deviation of the item.

`dmacs` The dMACS effect size, nonnegative.

The returned object carries four attributes: `"reference"` and `"focal"`, the two group labels, and `"mean_focal"` and `"sd_focal"`, the focal group's latent mean and standard deviation used in the integral (named by latent variable on the fit path).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Meredith, W. (1993). Measurement invariance, factor analysis and factorial invariance. *Psychometrika, 58*(4), 525–543. doi:10.1007/BF02294825

Millsap, R. E. (2011). *Statistical approaches to measurement invariance*. Routledge.

Nye, C. D., Bradburn, J., Olenick, J., Bialko, C., & Drasgow, F. (2019). How big are my effects? Examining the magnitude of effect sizes in studies of measurement equivalence. *Organizational Research Methods, 22*(3), 678–709. doi:10.1177/1094428118761122

Nye, C. D., & Drasgow, F. (2011). Effect size indices for analyses of measurement equivalence: Understanding the practical importance of differences between groups. *Journal of Applied Psychology, 96*(5), 966–980. doi:10.1037/a0022955

See Also

measurement_invariance for the invariance ladder that locates a violation; smd for the standardized mean difference whose pooling and scale dMACS borrows; cfa_1 for the single group measurement model.

Other multivariate and latent variable methods: average_variance_extracted(), bifactor_indices(), cfa_1(), cfa_2(), cfa_k(), ci_eigenvalue(), common_method_marker(), common_method_single_factor(), ecvi(), htmt(), irt_grm(), irt_information(), measurement_alignment(), measurement_invariance(), procrustes_phi(), simple_structure()

Examples

```
# Reported measurement equations: the two items share loadings, and the
# second item's intercept is 0.30 higher in the reference group. With no
# loading difference, dMACS is just 0.30 divided by the pooled SD.
dmacs(lambda_reference = c(0.80, 0.75), lambda_focal = c(0.80, 0.75),
      nu_reference = c(2.00, 2.30), nu_focal = c(2.00, 2.00),
      sd_pooled = c(1.20, 1.10), item_names = c("optimism", "worry"))

# A partial invariance model for the four spatial tests at the two
# Holzinger and Swineford schools. The anchors are constrained equal; the
# cubes and lozenges tests are freed, so only those two can move.
data(holzinger_swineford)
items <- c("t1_visual_perception", "t2_cubes",
           "t3_paper_form_board", "t4_lozenges")
model <- paste("spatial =~", paste(items, collapse = " + "))
fit <- lavaan::cfa(model, data = holzinger_swineford, group = "school",
                   group.equal = c("loadings", "intercepts"),
                   group.partial = c("spatial =~ t2_cubes", "t2_cubes ~ 1",
                                     "spatial =~ t4_lozenges",
                                     "t4_lozenges ~ 1"))
dmacs(fit)

# The broom verbs: one row per item, and the group metadata.
generics::tidy(dmacs(fit))
generics::glance(dmacs(fit))
```

---

dmar_output_helpers *Publication-Ready Display of DMAR Result Tables*

Description

The dmar_tbl print layer formats a result table for the console. These helpers carry the same formatting into the two places a researcher writes up an analysis: a knitted report and a results sentence.

Usage

```
## S3 method for class 'dmar_tbl'
knit_print(x, ...)

as_kable(x, ...)

## S3 method for class 'dmar_tbl'
as_kable(x, format = NULL, ...)

results_sentence(x, label = NULL, digits = 2)
```

Arguments

x	A dmar_tbl object (a data.frame returned by a DMAR function). For results_sentence, a dmar_tbl that carries a confidence interval, either as lower_limit / upper_limit rows of a long table or as lower_limit / upper_limit columns of a wide table.
...	Additional arguments. For knit_print.dmar_tbl and as_kable.dmar_tbl, passed to kable.
format	Passed to kable as its format argument (for example "html", "latex", "pipe"). The default NULL lets **knitr** choose based on the output context.
label	For results_sentence, the label that leads the sentence (for example "Cohen's d"). The default NULL uses the name of the point-estimate term.
digits	For results_sentence, the number of decimal places for the estimate and the interval limits. Default 2.

Details

knit_print.dmar_tbl is the **knitr** print method, so a dmar_tbl dropped into an R Markdown chunk renders as a formatted kable (sensible rounding, whole-number sample sizes, *p*-values to fixed decimals) rather than as a raw dump of doubles. as_kable is the explicit form of the same rendering: it returns the knitr_kable object so the caller can pipe it into further styling or embed it in a larger document. Both reuse format.dmar_tbl, so what a reader sees in a report matches what they saw at the console, and neither rounds the stored numbers.

results_sentence turns a table that carries a confidence interval into the one sentence an author puts in a results section, for example "smd = 0.50, 95% CI [0.10, 0.90]". It reads the estimate and its limits from the numeric columns at full precision and formats them for the sentence, so the reported numbers are exact to the requested decimals rather than transcribed from the rounded console display.

Value

knit_print.dmar_tbl returns a knit_asis object (the rendered table) for **knitr** to place in the document. as_kable returns a knitr_kable object. results_sentence returns a length-one character string.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

dmar_tbl for the console print layer and format_p for the *p*-value convention these helpers reuse.

Examples

```
x <- ci_smd(smd = 0.5, n_1 = 50, n_2 = 50)

# A knitr_kable that keeps every column, ready for a report.
as_kable(x)

# The sentence an author writes in a results section.
results_sentence(x, label = "Cohen's d")

# Wide tables (one interval per row) work the same way.
results_sentence(ci_R2(R2 = 0.25, N = 100, p = 5))
```

---

drinks_trial *Community Reinforcement Approach Drinking Trial With Homeless Alcohol-Dependent Individuals (Smith, Meyers, & Delaney, 1998)*

Description

Nine-month follow-up drinking outcomes for the $N = 88$ homeless alcohol-dependent participants in Smith, Meyers, and Delaney's (1998) randomized clinical trial of the Community Reinforcement Approach (CRA), published in the *Journal of Consulting and Clinical Psychology*. Participants were recruited from the Salvation Army Adult Rehabilitation Center in Albuquerque, New Mexico across two consecutive cohorts and were randomized to CRA, to CRA augmented with disulfiram, or to standard care. The outcome is the participant's average number of standard drinks per week at the nine-month follow-up, reported in both raw form (markedly right-skewed) and after a base-ten log transformation that approximately normalizes the distribution used in the original published analyses. The data are reproduced in Maxwell, Delaney, and Kelley (2027, *Designing Experiments and Analyzing Data: A Model Comparison Perspective*, 4th ed., Routledge), Chapter 3, Section 3.10.4, as the textbook's worked example of a between-subjects analysis of variance with a heavily skewed outcome.

Usage

```
drinks_trial
```

Format

A data frame with 88 observations on 5 variables.

`id` Sequential participant identifier, 1 to 88.

`cohort` Factor with levels 1 and 2. The study enrolled two consecutive cohorts. Cohort 1 compared three conditions (Standard, CRA, and CRA + Disulfiram); Cohort 2 dropped the disulfiram cell on the basis of Cohort 1 results and compared Standard against CRA only.

`treatment` Factor with levels `Standard`, `CRA`, and `CRA + Disulfiram`, the randomly assigned treatment condition.

`drinks_per_week` Average number of standard drinks per week at the nine-month follow-up. Bounded below at zero and markedly right-skewed (range 0 to 624.6, mean 36.9, median 3.8).

`log_drinks` Common-log transformation $\log_{10}(x + 1)$ of $x =$ `drinks_per_week`, the scale on which Smith, Meyers, and Delaney (1998) ran their primary between-groups analyses to obtain approximate normality. The plus-one inside the logarithm keeps the zero values finite (and mapped to zero).

Details

Per-cell sample sizes. The (`cohort`, `treatment`) crosstab is incomplete by design:

	Standard	CRA	CRA + Disulfiram
Cohort 1	17	15	19
Cohort 2	20	17	(not run)

Treatment marginals sum to 37 Standard, 32 CRA, and 19 CRA + Disulfiram; cohort marginals sum to 51 in Cohort 1 and 37 in Cohort 2.

The authors, the published article, and the textbook. The trial was conducted and reported by Jane Ellen Smith, Robert J. Meyers, and Harold D. Delaney, all then in the Department of Psychology at the University of New Mexico. Smith and Meyers were the substantive PIs of an extensive program of CRA research; Meyers (with N. H. Azrin) is widely associated with the dissemination of CRA and is the developer of the related Community Reinforcement and Family Training (CRAFT) intervention. Delaney is a quantitative psychologist who served as the trial's methodologist. In DMAR the data are included as a worked-example benchmark because they appear in Maxwell, Delaney, and Kelley (2027), Chapter 3, Section 3.10.4, as the running example for the consequences of skipping a normalizing transformation when fitting an analysis of variance to a markedly skewed outcome.

The original published article is

- Smith, J. E., Meyers, R. J., and Delaney, H. D. (1998). The community reinforcement approach with homeless alcohol-dependent individuals. *Journal of Consulting and Clinical Psychology, 66*(3), 541–548. doi:10.1037/0022006X.66.3.541

and the textbook reproduction (with Section 3.10.4 of MDK 2027 devoted to the worked analysis) is

- Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

The Community Reinforcement Approach (CRA). CRA is a behavioral treatment for alcohol-use disorder developed by Nathan Azrin and colleagues in the 1970s. It uses operant conditioning principles to align vocational, marital, social, and recreational reinforcers in the patient's natural environment with abstinence rather than drinking. CRA had previously been evaluated in housed

populations; Smith, Meyers, and Delaney (1998) extended its evidence base to homeless alcohol-dependent individuals, the population represented in these data.

Disulfiram (Antabuse). The CRA + Disulfiram condition added disulfiram, a long-standing pharmacological adjunct that produces an aversive reaction on alcohol consumption. The Cohort 1 finding that adding disulfiram offered little incremental benefit over CRA alone motivated dropping the disulfiram cell in Cohort 2.

Design and analysis. The published analyses fit a one-way analysis of variance to `log_drinks` (the raw `drinks_per_week` variable violates the normality assumption badly enough that the omnibus inference is misleading without transformation; see Maxwell, Delaney, and Kelley, 2027, Section 3.10.4). Useful contrasts include CRA versus Standard within each cohort, CRA versus CRA + Disulfiram within Cohort 1 to estimate the incremental benefit of disulfiram, and pooling across cohorts to estimate an overall CRA-versus-Standard effect.

Use as a DMAR benchmark. This data set is the canonical DMAR example for one-way between-subjects analysis with a heavily right-skewed continuous outcome. Methods that pair naturally with the data set include the one-way analysis of variance, planned contrasts (`contrast_test`), the standardized mean difference (`smd`, `ci_smd`), Cliff's delta (`cliff_delta`), the probability of superiority, and accuracy in parameter estimation sample size planning (`ss_aipe_smd`, `ss_power_smd`). The contrast between an ANOVA fit to `drinks_per_week` and one fit to `log_drinks` is a clean teaching example for the consequences of skipping a normalizing transformation.

What is and is not here. The published paper reports drinking outcomes at multiple follow-up timepoints (2, 6, 9, and 12 months) along with several demographic and clinical covariates. The values distributed here cover the nine-month follow-up only and contain no demographic or covariate information. Anyone wanting the full longitudinal trajectories or the covariate set should consult Smith, Meyers, and Delaney (1998) directly.

Author(s)

Ken Kelley

Source

Smith, J. E., Meyers, R. J., and Delaney, H. D. (1998). The community reinforcement approach with homeless alcohol-dependent individuals. *Journal of Consulting and Clinical Psychology, 66*(3), 541–548. doi:10.1037/0022006X.66.3.541

Also reproduced in Maxwell, Delaney, and Kelley (2027), Chapter 3.

The data are distributed openly with the companion materials of Maxwell, Delaney, and Kelley (2027), *Designing Experiments and Analyzing Data: A Model Comparison Perspective*, and are redistributed here on that basis.

References

Smith, J. E., Meyers, R. J., and Delaney, H. D. (1998). The community reinforcement approach with homeless alcohol-dependent individuals. *Journal of Consulting and Clinical Psychology, 66*(3), 541–548. doi:10.1037/0022006X.66.3.541

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on between-subjects analysis of variance and Section 3.10 on transformations.)

Kelley, K. (2026). *DMAR: Methods for the design, measurement, and analysis of human-centered outcomes in R* [R package]. https://github.com/yelleKneK/DMAR

Examples

```
data(drinks_trial)
str(drinks_trial)

# Per-cell sample sizes by cohort and treatment.
with(drinks_trial, table(cohort, treatment))

# Right skew of the raw outcome versus the log-transformed scale
# used in the published analyses.
summary(drinks_trial$drinks_per_week)
summary(drinks_trial$log_drinks)

# One-way analysis of variance on the log scale, treating the
# five filled (cohort, treatment) cells as the design.
fit <- aov(log_drinks ~ cohort:treatment, data = drinks_trial)
summary(fit)

# Standardized mean difference (Cohen's d) for every pairwise
# (two-way) comparison of the three treatments, on the normalizing
# log scale. The loop assembles a table of d with its 95%
# confidence interval for each of the three possible pairs.
groups <- split(drinks_trial$log_drinks, drinks_trial$treatment)
cmp    <- combn(names(groups), 2)
smd_table <- do.call(rbind, lapply(seq_len(ncol(cmp)), function(j) {
  g1 <- groups[[cmp[1, j]]]
  g2 <- groups[[cmp[2, j]]]
  d  <- smd(group_1 = g1, group_2 = g2)$value
  ci <- ci_smd(smd = d, n_1 = length(g1), n_2 = length(g2))
  data.frame(
    comparison = paste(cmp[1, j], "vs", cmp[2, j]),
    d          = round(d, 3),
    ci_lower   = round(ci$value[ci$term == "lower_limit"], 3),
    ci_upper   = round(ci$value[ci$term == "upper_limit"], 3)
  )
}))
smd_table
```

---

dunn_test *Provides Dunn's Rank-Sum Test of All Pairwise Differences Following a Kruskal–Wallis Test*

Description

Provides Dunn's Rank-Sum Test of All Pairwise Differences Following a Kruskal–Wallis Test

Usage

```
dunn_test(x, group = NULL, method = "holm")
```

Arguments

x	Either (a) a numeric vector of the outcome, in which case group must also be supplied, or (b) a fitted lm or aov object with a single factor predictor, from which the outcome and grouping factor are taken (the model itself is not used, since the test is distribution free).
group	When x is a vector, a factor (or coercible to factor) giving the group membership of each observation.
method	The multiplicity adjustment applied to the pairwise p-values, passed to p.adjust. Default "holm". Use "none" to obtain the unadjusted values.

Details

Which Dunn. Olive Jean Dunn published two different multiple comparison procedures, and both are called "Dunn's" in the literature. This function implements the *nonparametric* one of Dunn (1964), which compares mean ranks. It is not the Bonferroni procedure of Dunn (1961), which Maxwell, Delaney, and Kelley (2027, Chapter 5) call Dunn's procedure and which reaches DMAR through the method arguments of contrast_adjusted and p.adjust. The two are unrelated apart from their author, and the method argument here can apply the 1961 procedure to the 1964 procedure's p-values.

What it does. The Kruskal–Wallis test asks whether *any* of the groups differ; it does not say which. Dunn's test is its pairwise follow-up. All N observations are ranked together, with tied values receiving their average rank. Writing $\bar{R}_g$ for the mean rank of group g, each pair is compared with

$$z = \frac{\bar{R}_g - \bar{R}_h}{\sqrt{\left[\frac{N(N+1)}{12} - \frac{\sum_i (t_i^3 - t_i)}{12(N-1)}\right]\left(\frac{1}{n_g} + \frac{1}{n_h}\right)}},$$

where t_i is the number of observations tied at the ith distinct value; the second term in the brackets is the tie correction and vanishes when there are no ties. The statistic is referred to the standard normal distribution, and the resulting p-values are adjusted for multiplicity by method.

The pooled ranking is what makes this the right follow-up. Dunn's test ranks across all groups at once and uses the variance of the ranks implied by the Kruskal–Wallis null, so it is consistent with the omnibus test that preceded it. Running a separate Mann–Whitney test on each pair instead re-ranks the data within every pair, which answers a different question for every comparison and is not coherent with the omnibus result.

When to use it. Use Dunn's test when the outcome is ordinal, or when it is continuous but the normality or homogeneity assumptions behind ci_tukey_kramer and ci_games_howell are untenable and a rank-based analysis is preferred to a transformation; the design is between subjects; and a significant Kruskal–Wallis test leaves the question of which groups differ. It is the rank analogue of Tukey's HSD, in the sense of covering all $a(a-1)/2$ pairs.

What it does not tell you. The test compares mean *ranks*, not medians. A significant pair means one group's observations tend to be larger, that is, stochastic dominance; it does not by itself license a statement about medians unless the group distributions have the same shape. It also returns no

confidence interval on any quantity in the original units, which is a real cost: where a parametric procedure is defensible, `ci_games_howell` or `ci_tukey_kramer` reports intervals on the mean difference, and Maxwell, Delaney, and Kelley (2027) emphasize interval estimation over test decisions throughout. For a distribution free effect size with a confidence interval, see `cliff_delta`.

Value

A `data.frame` with one row per pairwise comparison and columns `contrast`, `mean_rank_difference`, `se`, `z_statistic`, `p_value`, and `p_adjusted`. The table prints through the `dmar_tbl` display layer.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Dunn, O. J. (1964). Multiple comparisons using rank sums. *Technometrics, 6*(3), 241–252. doi:10.1080/00401706.1964.10490181

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association, 56*(293), 52–64. doi:10.2307/2282330 (The Bonferroni procedure; not what this function computes.)

Kruskal, W. H., & Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association, 47*(260), 583–621. doi:10.2307/2280779

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

`kruskal.test` for the omnibus test this follows, `ci_games_howell` and `ci_tukey_kramer` for the parametric all-pairs procedures, `cliff_delta` for a distribution free effect size with a confidence interval, and `p.adjust` for the `method` choices.

Examples

```
# The omnibus question first: do the three treatment arms of the
# drinks_trial data differ at all? The raw drinks_per_week outcome is
# markedly right-skewed, which is no obstacle here: ranks are invariant
# to a monotone transformation, so the raw and log scales give
# identical results.
kruskal.test(drinks_per_week ~ treatment, data = drinks_trial)

# Then which pairs, on the same pooled ranking the omnibus test used.
dunn_test(drinks_trial$drinks_per_week, drinks_trial$treatment)

# Without a multiplicity adjustment (rarely what you want).
dunn_test(drinks_trial$drinks_per_week, drinks_trial$treatment,
          method = "none")
```

`ecvi` *Expected Cross-Validation Index (ECVI) for a Covariance-Structure Model*

Description

The ECVI of Browne and Cudeck (1989) estimates how well a fitted model's implied covariance matrix would fit an independent sample of the same size from the same population. It is the single-sample estimate of the cross-validation discrepancy, so a smaller ECVI indicates a model expected to generalize better; ECVI is most useful for comparing competing models fit to the same data. A confidence interval, derived from the noncentral chi square distribution, accompanies the point estimate.

Usage

```
ecvi(
  fit = NULL,
  chisq = NULL,
  df = NULL,
  npar = NULL,
  n = NULL,
  conf_level = 0.95
)
```

Arguments

`fit`	A fitted **lavaan** model. Supply this, or the summary statistics below.
`chisq`, `df`, `npar`, `n`	The model chi square, its degrees of freedom, the number of free parameters, and the total sample size. Used when `fit` is not supplied, so an ECVI can be obtained from a published fit table.
`conf_level`	Confidence level for the interval. Defaults to 0.95.

Details

With q free parameters and total sample size N, ECVI $= (\chi^2 + 2q)/N$, the value the *Journal of Statistical Software* reference implementation in **lavaan** reports. Writing $\hat{\lambda} = \chi^2 - df$ for the estimated noncentrality, this is $(\hat{\lambda} + df + 2q)/N$; the confidence interval replaces $\hat{\lambda}$ by the lower and upper noncentrality limits from `ci_nc_chisq`, the same inversion used for the RMSEA interval (see `ci_rmsea`). ECVI differs from the AIC only by the constant factor N, so the two rank models identically; ECVI is reported because its metric (a discrepancy per observation) and its confidence interval are interpretable on their own.

Value

A `data.frame` (class `dmar_tbl`) with rows `ecvi`, `lower_limit`, and `upper_limit` in the `value` column.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Browne, M. W., & Cudeck, R. (1989). Single sample cross-validation indices for covariance structures. *Multivariate Behavioral Research, 24*(4), 445–455.

See Also

ci_rmsea, ci_nc_chisq.

Other multivariate and latent variable methods: average_variance_extracted(), bifactor_indices(), cfa_1(), cfa_2(), cfa_k(), ci_eigenvalue(), common_method_marker(), common_method_single_factor(), dmacs(), htmt(), irt_grm(), irt_information(), measurement_alignment(), measurement_invariance(), procrustes_phi(), simple_structure()

Examples

```
# From a published fit table (no model object needed).
ecvi(chisq = 24.361, df = 8, npar = 13, n = 301)

fit <- lavaan::cfa(
  "visual =~ t1_visual_perception + t2_cubes + t4_lozenges
   verbal =~ t6_paragraph_comprehension + t7_sentence + t9_word_meaning",
  data = holzinger_swineford, std.lv = TRUE)
ecvi(fit)
```

---

effects_coding *Effects-Coding Contrast Matrix for a Factor*

Description

Builds the effects-coding (also called *deviation coding* or *sum-to-zero*) contrast matrix for a factor with a levels. Each non-reference level contrasts with the grand mean (rather than with a reference category as in dummy coding). The returned matrix has rows = levels and columns named after the levels, replacing the numeric column names produced by `stats::contr.sum()`.

Usage

```
effects_coding(levels, reference = NULL)
```

Arguments

`levels`	Either an integer giving the number of levels or a character / factor vector giving the level labels. If integer, the labels default to `"L1"`, `"L2"`, ...
`reference`	Optional character name of the reference level (whose coefficients are all -1). Default is the last level.

Details

Why effects coding. Effects coding gives the regression intercept the interpretation of the grand mean (rather than the reference-category mean), and each slope coefficient becomes the deviation of that level's mean from the grand mean (rather than the difference vs the reference category). For balanced designs the effect coefficients are orthogonal to the intercept.

Equivalent to. `stats::contr.sum()` but with meaningful column names (the level labels), which is what is lost in the base-R implementation.

Value

A numeric $a \times (a-1)$ matrix with row names = the factor levels and column names = the non-reference levels. Suitable for assignment to `contrasts(factor)` or use in manual contrast construction.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (3rd ed.). Lawrence Erlbaum. (See Chapter 8.)

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapters 4, 7.)

See Also

`contr.sum`, `helmert_coding`, `is_orthogonal_set`

Other design utilities: `design_consequences()`, `design_effect()`, `helmert_coding()`, `is_orthogonal_set()`, `orthogonal_polynomial()`

Examples

```
# 1. Effects coding for a 4-level factor:
effects_coding(c("low", "med", "high", "very_high"))

# 2. With "low" as the reference category:
effects_coding(c("low", "med", "high", "very_high"),
               reference = "low")

# 3. Assign to a factor for modeling:
f <- factor(c("a", "b", "c", "a", "b", "c"))
contrasts(f) <- effects_coding(levels(f))
model.matrix(~ f)
```

`epsilon_corrections` *Greenhouse-Geisser, Huynh-Feldt, and Lower-Bound Epsilon Corrections*

Description

Computes the three standard sphericity-correction factors for the univariate within-subjects *F* test. When sphericity holds, all three equal 1; departures from sphericity reduce them, deflating the effective degrees of freedom and thus tempering the inflated Type I error rate of the unadjusted univariate test.

Usage

```
epsilon_corrections(x, id = NULL, time = NULL, outcome = NULL)
```

Arguments

`x`	Either an $n \times k$ numeric matrix or `data.frame` (rows = subjects, columns = repeated measurements); *or* a long-format `data.frame` together with `id`, `time`, and `outcome` column names.
`id`	Column name in `x` identifying the subject when `x` is in long format (`NULL` otherwise).
`time`	Column name in `x` identifying the within-subjects factor level when `x` is in long format (`NULL` otherwise).
`outcome`	Column name in `x` identifying the dependent variable when `x` is in long format (`NULL` otherwise).

Details

For an $(k-1) \times (k-1)$ covariance matrix $\hat{\Sigma}_C$ of orthonormal contrasts among the k repeated measurements (with eigenvalues $\lambda_1, \ldots, \lambda_{k-1}$):

$$\hat{\varepsilon}_{\mathrm{GG}} = \frac{(\sum \lambda_i)^2}{(k-1)\sum \lambda_i^2},$$

$$\hat{\varepsilon}_{\mathrm{HF}} = \min\Big(1,\ \frac{n(k-1)\hat{\varepsilon}_{\mathrm{GG}} - 2}{(k-1)\big(n-1-(k-1)\hat{\varepsilon}_{\mathrm{GG}}\big)}\Big),$$

$$\hat{\varepsilon}_{\mathrm{LB}} = \frac{1}{k-1}.$$

The Greenhouse-Geisser $\hat{\varepsilon}$ tends to be conservative; the Huynh-Feldt correction adjusts upward to be (approximately) unbiased; the lower bound is the worst-case adjustment.

Value

A `data.frame` with columns `epsilon_method` (`"Greenhouse-Geisser"`, `"Huynh-Feldt"`, `"lower_bound"`) and `epsilon` (the correction factor in $[1/(k-1), 1]$).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Greenhouse, S. W., & Geisser, S. (1959). On methods in the analysis of profile data. *Psychometrika, 24*(2), 95–112.

Huynh, H., & Feldt, L. S. (1976). Estimation of the Box correction for degrees of freedom from sample data in randomized block and split-plot designs. *Journal of Educational Statistics, 1*(1), 69–82.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

mauchly_test, anova_within

Other within-subjects analysis: anova_within(), anova_within_two_way(), mauchly_test(), pairwise_within(), plot_trajectories_fitted()

Examples

```
set.seed(113)
Y <- matrix(rnorm(20 * 4), nrow = 20)
epsilon_corrections(Y)
```

---

equivalence_c *Equivalence and Noninferiority Tests for a Linear Contrast via Two One-Sided Tests (TOST)*

Description

Performs the two one-sided tests procedure (Schuirmann, 1987) for equivalence, and the companion one-sided noninferiority test, for a linear contrast of group means $\psi = \sum_j c_j \mu_j$ in a fixed effects design with one pooled error term. The equivalence null hypothesis is that the population contrast lies outside the user-specified bounds, $H_0 : \psi \leq -\delta_L \cup \psi \geq \delta_U$, against the alternative $H_1 : -\delta_L < \psi < \delta_U$. The noninferiority null is $H_0 : \psi \leq -\delta_L$ against $H_1 : \psi > -\delta_L$. Following Chattopadhyay, Bandyopadhyay, Kelley, and Padalunkal (2025), the equivalence decision is read from the 100(1 - 2α)% confidence interval: equivalence is declared when the whole interval lies inside $(-\delta_L, \delta_U)$, and noninferiority when the interval's lower limit exceeds $-\delta_L$.

Usage

```
equivalence_c(
  means = NULL,
  s_anova = NULL,
  c_weights = NULL,
  n = NULL,
  psi_hat = NULL,
  se = NULL,
  df_error = NULL,
  delta_lower = NULL,
  delta_upper = NULL,
  benchmark = NULL,
  alpha_level = 0.05
)
```

Arguments

means	A vector of group means. Supply together with s_anova, c_weights, and n. Alternatively, use the direct interface via psi_hat, se, and df_error.
s_anova	The standard deviation of the errors from the ANOVA model (the square root of the mean square error), as in ci_c.
c_weights	The contrast weights. For a mean comparison the weights must sum to zero, and, so that the bounds are on the raw scale of the response, the positive weights must sum to 1 and the negative weights to -1 (use fractional values, not integers). When benchmark is supplied, the weights must instead be nonnegative and sum to 1 (typically a single 1 selecting one group).
n	Sample sizes per group (if length 1, equal group sizes are assumed).
psi_hat	The estimated contrast, for the direct interface. Supply together with se and df_error when the contrast and its standard error have already been computed (for example, from a fitted model with covariates).
se	The standard error of psi_hat, for the direct interface.
df_error	The error degrees of freedom. On the summary-statistic interface the default is $N - J$, with J the number of groups; it must be supplied for designs with additional factors. Required on the direct interface.
delta_lower, delta_upper	Equivalence bounds on the raw scale of the response. Both must be positive; the equivalence region is $(-\delta_L, +\delta_U)$. If only delta_upper is supplied, the bounds are symmetric. Noninferiority uses $-\delta_L$ alone.
benchmark	An optional known constant to compare against (for example, a normative or regulatory cutoff). When supplied, the contrast is $\sum_j c_j \mu_j - b$ with nonnegative weights summing to 1, and the constant contributes no sampling variability.
alpha_level	One-sided significance level for each of the two tests. Default 0.05, so the interval the decisions are read from is the 90% CI.

Details

One pooled error term. On the summary-statistic interface the standard error is $\mathrm{SE}(\hat{\psi}) = s_{\mathrm{anova}}\sqrt{\sum_j c_j^2/n_j}$, the model comparison position of Maxwell, Delaney, and Kelley (2027): every one-degree-of-freedom contrast is judged against the same yardstick, the root mean square error of one model fit to all groups.

The verdict logic. Reading the 100(1 - 2α)% CI against the bounds: an interval entirely inside $(-\delta_L, \delta_U)$ is *equivalent*; entirely above δ_U is *superior* (which implies noninferior); entirely below $-\delta_L$ is *inferior*; a lower limit above $-\delta_L$ with an upper limit past δ_U is *noninferior only*; and an interval straddling a bound is *inconclusive*. An inconclusive result is a statement about precision, not evidence of a difference: only an interval clearing a bound entirely licenses a directional claim.

Why the weights must sum to ± 1. A bound stated in raw units of the response is only meaningful if the contrast is itself a simple difference of (weighted) means on that scale, which requires the positive weights to sum to 1 and the negative weights to -1. A weight vector such as `c(2, -2)` would silently double the effective bounds, so it is rejected rather than rescaled.

Choosing the bounds. The bounds must be fixed before the data are examined, on substantive grounds: the smallest difference that would matter (Serlin & Lapsley, 1985; Lakens, Scheel, & Isager, 2018). They are never derived from a standard error, which would make the definition of "close enough" a function of the sample size.

Agreement with emmeans. The *p*-values reproduce `emmeans::test(..., side = "equivalence")` and `emmeans::test(..., side = "noninferiority")` with `adjust = "none"` to machine precision. Note two emmeans pitfalls the interface here avoids: `side = "left"` tests non-superiority, not noninferiority, and `trt.vs.ctrl` families silently apply a Dunnett-type adjustment unless `adjust = "none"` is passed.

Value

A `data.frame` with rows for the estimated contrast (`psi_hat`), its standard error (`se`), the error degrees of freedom (`df`), the two one-sided test statistics (`t_lower`, `t_upper`) and their *p*-values (`p_lower`, `p_upper`), the joint TOST *p*-value (`p_tost`, the larger of the two), the noninferiority *p*-value (`p_noninferiority`, equal to `p_lower` by construction), the 100(1 - 2α)% confidence limits (`lower_limit`, `upper_limit`), the bounds (`delta_lower`, stored as the signed lower bound, and `delta_upper`), and four binary decision flags (`equivalent`, `noninferior`, `superior`, `inferior`; 1 = declared, 0 = not). When all four flags are 0, the interval straddles a bound and the result is inconclusive. The five-way classification is also attached as the `"verdict"` attribute, one of `"Equivalent"`, `"Superior"`, `"Inferior"`, `"Noninferior only"`, or `"Inconclusive"`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Chattopadhyay, B., Bandyopadhyay, T., Kelley, K., & Padalunkal, J. J. (2025). A sequential approach for noninferiority or equivalence of a linear contrast under cost constraints. *Psychological Methods, 30*(2), 425–439. doi:10.1037/met0000570

Lakens, D., Scheel, A. M., & Isager, P. M. (2018). Equivalence testing for psychological research: A tutorial. *Advances in Methods and Practices in Psychological Science, 1*(2), 259–269. doi:10.1177/2515245918770963

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means.)

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics, 15*(6), 657–680.

Serlin, R. C., & Lapsley, D. K. (1985). Rationality in psychological research: The good-enough principle. *American Psychologist, 40*(1), 73–83.

Wellek, S. (2010). *Testing statistical hypotheses of equivalence and noninferiority* (2nd ed.). Chapman & Hall/CRC.

See Also

equivalence_smd, equivalence_r, ci_c, contrast_test, power_equivalence_c, ss_power_equivalence_c, plot_equivalence

Other equivalence testing: equivalence_r(), equivalence_smd(), plot_equivalence(), power_density_equivalence, power_equivalence_c(), power_equivalence_md(), power_equivalence_md_plot(), ss_power_equivalence_c()

Examples

```
# 1. Two of five groups compared against the reference group, with a
#    pooled error term from one model across all five groups.
#    Bounds of 5 raw-scale points; alpha_level = .05, so decisions read
#    from the 90% CI.
equivalence_c(means = c(70.40, 55.61, 51.91, 65.66, 65.12),
       s_anova = 15.67,
       c_weights = c(-1, 0, 0, 0, 1),
       n = c(113, 74, 76, 80, 61),
       delta_upper = 5)

# 2. The same contrast through the direct interface, as when the
#    estimate and standard error come from a model with covariates.
res <- equivalence_c(psi_hat = -5.28, se = 2.49, df_error = 399,
              delta_upper = 5)
res
attr(res, "verdict")

# 3. A group mean against a fixed benchmark of 68: the constant
#    contributes no sampling variability.
equivalence_c(means = c(70.40, 55.61, 51.91, 65.66, 65.12),
       s_anova = 15.67,
       c_weights = c(1, 0, 0, 0, 0),
       n = c(113, 74, 76, 80, 61),
       benchmark = 68, delta_upper = 5)

# 4. Asymmetric bounds: a shortfall of 3 matters, an excess of 8 does.
equivalence_c(psi_hat = 1.2, se = 1.1, df_error = 120,
```

```
        delta_lower = 3, delta_upper = 8)
```

`equivalence_r` *Equivalence Test for the Pearson Correlation via Two One-Sided Tests (TOST)*

Description

Performs a two one-sided tests procedure for equivalence of a Pearson correlation ρ to zero against user-specified equivalence bounds $[-\rho_L, \rho_U]$ (Counsell & Cribbie, 2015; Goertzen & Cribbie, 2010). Uses the Fisher's Z transformation throughout, a large-sample approximation whose accuracy under bivariate normality improves quickly with *n*. Equivalence is declared when the 100(1 - 2α)% Fisher's Z CI on ρ lies entirely inside the equivalence region.

Usage

```
equivalence_r(
  r = NULL,
  n = NULL,
  x = NULL,
  y = NULL,
  rho_lower = NULL,
  rho_upper = NULL,
  alpha_level = 0.05
)
```

Arguments

`r`, `n`	Observed sample correlation *r* and sample size. Alternatively supply `x` and `y` to compute *r* from raw data.
`x`, `y`	Numeric vectors of paired observations. If supplied, `r` and `n` are computed from the data and the `r`/`n` arguments are ignored.
`rho_lower`, `rho_upper`	Equivalence bounds on the correlation scale, both positive. The equivalence region is $[-\rho_L, +\rho_U]$. If only `rho_upper` is supplied, the bounds are symmetric.
`alpha_level`	One-sided significance level. Default `0.05`.

Details

Fisher's Z transformation.

$$Z = \tfrac{1}{2}\log\left(\frac{1+r}{1-r}\right), \quad \mathrm{Var}(Z) = \frac{1}{n-3}.$$

The TOST is run on the Fisher's Z scale:

- Lower test: $(Z - Z_{-\rho_L})\sqrt{n-3}$ compared against the upper α of $N(0,1)$.

- Upper test: $(Z - Z_{\rho_U})\sqrt{n-3}$ compared against the lower α of $N(0,1)$.

The CI bounds are back-transformed from the Z scale via $r = \tanh(Z)$ so they remain in $[-1, 1]$.

Choosing ρ_L and ρ_U. Common choices in psychology are 0.1, or domain-specific meaningfulness thresholds (e.g., 0.2 for cognitive task correlations). The bounds must be set *before* data collection.

Value

A `data.frame` with rows for the observed *r*, the two one-sided test statistics on the Fisher's Z scale, their *p*-values, the joint TOST *p*-value, the 100(1 - 2α)% CI on ρ, the equivalence bounds, a binary equivalence flag, and the sample size (`n`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Counsell, A., & Cribbie, R. A. (2015). Equivalence tests for comparing correlation and regression coefficients. *British Journal of Mathematical and Statistical Psychology, 68*(2), 292–309. doi:10.1111/bmsp.12045

Goertzen, J. R., & Cribbie, R. A. (2010). Detecting a lack of association: An equivalence testing approach. *British Journal of Mathematical and Statistical Psychology, 63*(3), 527–537. doi:10.1348/000711009X475853

Lakens, D. (2017). Equivalence tests: A practical primer for *t* tests, correlations, and meta-analyses. *Social Psychological and Personality Science, 8*(4), 355–362. doi:10.1177/1948550617697177

See Also

`equivalence_smd`, `ci_r`

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `correlations_test()`, `equivalence_smd()`, `factorial_anova()`, `manova_split_plot()`, `mauchly_test()`, `mixed_anova()`, `obrien_test()`, `pairwise_within()`, `randomization_test()`, `randomization_test_paired()`, `regions_of_significance()`, `simple_effects_AB()`, `summary_t_test()`, `welch_t()`

Other equivalence testing: `equivalence_c()`, `equivalence_smd()`, `plot_equivalence()`, `power_density_equivalence`
`power_equivalence_c()`, `power_equivalence_md()`, `power_equivalence_md_plot()`, `ss_power_equivalence_c()`

Examples

```
# 1. Equivalence test that |rho| < 0.10 with n = 200 and r = 0.05:
equivalence_r(r = 0.05, n = 200, rho_upper = 0.10)

# 2. From raw data:
set.seed(113)
x <- rnorm(150); y <- 0.04 * x + rnorm(150)
equivalence_r(x = x, y = y, rho_upper = 0.15)
```

`equivalence_smd` *Equivalence Test for the Standardized Mean Difference via Two One-Sided Tests (TOST)*

Description

Performs a two one-sided tests procedure (Schuirmann, 1987) for equivalence between two independent groups on the standardized mean difference (Cohen's *d*) scale. The null hypothesis is that the true δ lies outside the user-specified equivalence bounds $[-\delta_L, \delta_U]$; the alternative is that δ lies inside them. The test is the joint pair of one-sided *t* tests: $H_{0,L} : \delta \leq -\delta_L$ vs $H_{1,L} : \delta > -\delta_L$, and $H_{0,U} : \delta \geq \delta_U$ vs $H_{1,U} : \delta < \delta_U$. Equivalence is declared when *both* null hypotheses are rejected at level α.

Usage

```
equivalence_smd(
  x = NULL,
  y = NULL,
  smd = NULL,
  n_1 = NULL,
  n_2 = NULL,
  delta_lower = NULL,
  delta_upper = NULL,
  alpha_level = 0.05
)
```

Arguments

`x`, `y`	Numeric vectors of observations from the two groups. Alternatively, supply `smd`, `n_1`, `n_2` via the summary-statistic interface.
`smd`	Observed standardized mean difference (Cohen's *d*). Required if `x` and `y` are not supplied.
`n_1`, `n_2`	Group sample sizes. Required if `x` and `y` are not supplied.
`delta_lower`, `delta_upper`	Equivalence bounds on the *d* scale. Both must be positive; the equivalence region is $[-\delta_L, +\delta_U]$. If only `delta_upper` is supplied, the bounds are symmetric: $[-\delta_U, +\delta_U]$.
`alpha_level`	One-sided significance level for each of the two tests. Default `0.05`. The TOST is then equivalent to a 100(1 - 2α)% CI on δ lying inside the equivalence bounds.

Details

Schuirmann's TOST. The TOST procedure tests $H_0 : \delta \leq -\delta_L \cup \delta \geq \delta_U$ against $H_1 : -\delta_L < \delta < \delta_U$. Both component tests are rejected (and equivalence is declared) when the 100(1 - 2α)% CI on δ lies entirely inside $[-\delta_L, \delta_U]$.

Critical insight. A non-significant conventional NHST (t test of $\delta = 0$) is *not* evidence of equivalence; it only means we cannot reject $\delta = 0$. TOST inverts the testing logic so that "no meaningful effect" is the alternative, not the null.

Choosing δ_L and δ_U. The equivalence bounds must be set *before* data collection, based on what constitutes the smallest effect size of practical interest (Lakens, Scheel, & Isager, 2018). Common choices in the literature are 0.2 or 0.3, but the bound should reflect domain-specific meaningfulness.

Connection to the CI. The TOST rejection at level α is equivalent to the 100(1 - 2α)% Cohen's-*d* CI (i.e., 90% for the default $\alpha = 0.05$) lying entirely inside the equivalence region. This is the "two-one-sided" equivalence and matches the Westlake (1972) rationale for symmetric bioequivalence CIs.

Standard error and approximation. Each one-sided component is a Wald t test, $t = (\hat{d} \mp \delta)/\mathrm{SE}(\hat{d})$, referred to a central t distribution on $n_1 + n_2 - 2$ degrees of freedom, with the Hedges and Olkin (1985) large-sample standard error $\mathrm{SE}(\hat{d}) = \sqrt{(n_1 + n_2)/(n_1 n_2) + \hat{d}^2/(2(n_1 + n_2))}$. This is the asymptotic form of Schuirmann's (1987) two one-sided tests applied on the standardized scale; it is accurate in moderate-to-large samples but is an approximation to the exact noncentral t inversion used by `ci_smd`, and the two can differ in small samples.

Value

A `data.frame` with rows for the observed *d*, the two one-sided test statistics (`t_lower`, `t_upper`) and their degrees of freedom (`df`), the two one-sided *p*-values (`p_lower`, `p_upper`), the joint TOST *p*-value (the larger of the two), the 100(1 - 2α)% CI on δ, the equivalence bounds, and a binary decision flag (`equivalent`: 1 = equivalent, 0 = not).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. Academic Press.

Lakens, D. (2017). Equivalence tests: A practical primer for *t* tests, correlations, and meta-analyses. *Social Psychological and Personality Science, 8*(4), 355–362. doi:10.1177/1948550617697177

Lakens, D., Scheel, A. M., & Isager, P. M. (2018). Equivalence testing for psychological research: A tutorial. *Advances in Methods and Practices in Psychological Science, 1*(2), 259–269. doi:10.1177/2515245918770963

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics, 15*(6), 657–680.

Westlake, W. J. (1972). Use of confidence intervals in analysis of comparative bioavailability trials. *Journal of Pharmaceutical Sciences, 61*(8), 1340–1341.

See Also

`equivalence_r`, `ss_aipe_equivalence_smd`, `ci_smd`, `smd`

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `correlations_test()`,

equivalence_r(), factorial_anova(), manova_split_plot(), mauchly_test(), mixed_anova(), obrien_test(), pairwise_within(), randomization_test(), randomization_test_paired(), regions_of_significance(), simple_effects_AB(), summary_t_test(), welch_t()

Other equivalence testing: equivalence_c(), equivalence_r(), plot_equivalence(), power_density_equivalence_m power_equivalence_c(), power_equivalence_md(), power_equivalence_md_plot(), ss_power_equivalence_c()

Examples

```
# 1. Two groups, raw data, equivalence bound delta = 0.4:
set.seed(113)
x <- rnorm(50, 100, 15); y <- rnorm(50, 101, 15)
equivalence_smd(x = x, y = y, delta_upper = 0.4)

# 2. Summary-statistic interface: published d = 0.05, n = 40 per group.
equivalence_smd(smd = 0.05, n_1 = 40, n_2 = 40, delta_upper = 0.3)

# 3. Asymmetric bounds (acceptable -0.2 to +0.5):
equivalence_smd(smd = 0.10, n_1 = 80, n_2 = 80,
         delta_lower = 0.2, delta_upper = 0.5)
```

---

eta_squared *Eta Squared (Effect Size for ANOVA)*

Description

Computes the sample eta squared (η^2), the proportion of variance in the dependent variable accounted for by a fixed effect. Accepts either the raw ANOVA summary (F and the effect and error degrees of freedom) or a fitted model object. Supports both between-subjects designs (single-stratum aov or lm fits) and within-subjects / mixed designs (aovlist fits produced by aov with an Error() term in the formula). For factorial and within-subjects designs the function returns *partial* η^2 per effect (one row per non-Residuals effect across all strata), each computed against its own stratum's error term.

Usage

```
eta_squared(object = NULL, F_value = NULL, df_effect = NULL, df_error = NULL)
```

Arguments

object	Optional. A fitted model object of class aov, lm, or aovlist (a multi-stratum aov fit such as aov(y ~ A + Error(subject/A), data = d)). When supplied, the function loops over the non-Residuals effects across all error strata and returns one row per effect, with the stratum reported.
F_value	Observed F-value from the fixed-effects ANOVA (ignored if object is supplied).
df_effect	Numerator degrees of freedom for the effect (ignored if object is supplied).
df_error	Error (residual) degrees of freedom (ignored if object is supplied).

Details

The confidence interval is provided by the separate `ci_eta_squared`, paralleling the existing `smd`/`ci_smd` pairing.

Point estimate. The function uses the algebraically equivalent F-and-df form

$$\hat{\eta}^2 = \frac{df_{\text{effect}} \cdot F}{df_{\text{effect}} \cdot F + df_{\text{error}}},$$

which equals $SS_{\text{effect}}/(SS_{\text{effect}} + SS_{\text{error}})$. In a one-way ANOVA this is also $SS_{\text{effect}}/SS_{\text{total}}$, the conventional *total* η^2. In a factorial design the same expression yields *partial* η^2 for each effect, because SS_{error} appears in the denominator instead of SS_{total}; this matches the convention used by `ci_omega_squared`.

Designs supported.

- *Between-subjects ANOVA* (one-way or factorial): supply either a fitted `aov`/`lm` model or the raw F and degrees of freedom for a single effect.
- *Within-subjects or mixed ANOVA*: supply a fitted `aovlist` model produced with an `Error()` term, e.g.\ `aov(y ~ time + Error(subject/time), data = d)`. The function walks every error stratum returned by `summary(object)` and reports each effect with the stratum's residual *df*, so the η^2 value uses the stratum's specific error term. The reported `stratum` column tells you which one.

For more advanced model classes (`lmerMod`, `lme`, etc.) the fitted-model interface is not yet supported; supply the relevant F and degrees of freedom via the raw interface.

Sums of squares in factorial designs. `anova()` on an `aov`/`lm` uses Type I (sequential) sums of squares. For balanced designs all three types agree; for unbalanced designs they differ. If Type II or III F-values are required, compute them with e.g.\ `car::Anova(object, type = 3)` and pass the relevant F and degrees of freedom into the raw-argument interface.

Generalized eta squared, comparable across designs. The basic η^2 (and partial η^2) returned here are not comparable across studies that differ in factor structure. See `eta_squared_generalized` for a comparable alternative (Olejnik & Algina, 2003; Bakeman, 2005).

Value

A `data.frame` with one row per effect. For single-stratum (`aov`/`lm`) fits and the raw-argument interface the columns are `effect`, `eta_squared`, `F_value`, `df_effect`, `df_error`. For multi-stratum (`aovlist`) fits an additional `stratum` column reports which error stratum each effect's F test came from. When the raw-argument interface is used, `effect` is `"overall"`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bakeman, R. (2005). Recommended effect size statistics for repeated measures designs. *Behavior Research Methods, 37*(3), 379–384. doi:10.3758/BF03192707

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*, 137–152. doi:10.1037/a0028086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on η^2, Chapter 7 on factorial designs, and Chapter 11 on generalized η^2 for within-subjects designs.)

Olejnik, S., & Algina, J. (2003). Generalized eta and omega squared statistics: Measures of effect size for some common research designs. *Psychological Methods, 8*(4), 434–447. doi:10.1037/1082989X.8.4.434

Steiger, J. H. (2004). Beyond the *F* test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

ci_eta_squared, eta_squared_partial, ci_omega_squared, ci_pvaf

Other effect size estimates: cles(), cliff_delta(), correction_for_attenuation(), eta_squared_generalized(), eta_squared_partial(), expected_partial_r(), expected_r(), expected_smd(), nnt_from_smd(), omega_squared(), omega_squared_partial(), probability_of_superiority_paired(), proportion_of_superiorit responder_analysis(), smd_trimmed()

Examples

```
# 1. Raw-argument interface. Bargman's (1970) 5-group one-way ANOVA,
#        also used in Venables (1975), Fleishman (1980), and Steiger (2004):
#        11 subjects per group, observed F = 11.221.
eta_squared(F_value = 11.221, df_effect = 4, df_error = 50)

# 2. One way ANOVA from a fitted model (depression_bdi: three
#        treatment arms, 10 per arm, N = 30).
fit_one <- aov(bdi_post ~ condition, data = depression_bdi)
eta_squared(fit_one)

# 3. Factorial ANOVA: partial eta squared per effect (pygmalion
#        data: expectancy treatment x grade, 2 x 6 with unequal cell
#        sizes, N = 310). The treatment is manipulated; grade is a
#        measured classification of the pupils.
fit_factorial <- aov(iq_8 ~ treatment * factor(grade), data = pygmalion)
eta_squared(fit_factorial)

# 4. Within-subjects (repeated measures) ANOVA. Simulated 20-subject
#        x 3-time design; each subject is measured at Pre, Mid, Post.
set.seed(113)
n <- 20
rm_data <- data.frame(
  subject = factor(rep(seq_len(n), each = 3)),
```

```
  time    = factor(rep(c("Pre", "Mid", "Post"), n),
                   levels = c("Pre", "Mid", "Post")),
  y       = rnorm(n, sd = 1.5)[rep(seq_len(n), each = 3)] +
            0.7 * rep(1:3, n) + rnorm(n * 3, sd = 1.2)
)
fit_rm <- aov(y ~ time + Error(subject/time), data = rm_data)
eta_squared(fit_rm)   # 'stratum' column identifies the within-subjects error
```

`eta_squared_generalized` *Generalized Eta Squared (Effect Size for ANOVA, Comparable Across Designs)*

Description

Computes the sample generalized eta squared (η^2_G; Olejnik & Algina, 2003; Bakeman, 2005), the proportion of variance in the dependent variable accounted for by a fixed effect after the variance attributable to *measured* (observed) factors, but not other *manipulated* factors, has been left in the denominator. This makes η^2_G comparable across designs that differ in which factors are present, which regular η^2 and partial η^2 are not.

Usage

```
eta_squared_generalized(
  object = NULL,
  observed = NULL,
  SS_effect = NULL,
  SS_observed = NULL,
  SS_error = NULL,
  F_effect = NULL,
  df_effect = NULL,
  F_observed = NULL,
  df_observed = NULL,
  df_error = NULL
)
```

Arguments

`object`	Optional. A fitted model object of class `aov`, `lm`, or `aovlist` (multi-stratum aov fit, e.g.\ `aov(y ~ A + Error(subject/A), data = d)`).
`observed`	Character vector naming the *measured* (rather than *manipulated*) factors. Their sums of squares, and the SS of every interaction containing a listed factor, are kept in the denominator of η^2_G per Olejnik and Algina (2003, Eq. 5). Manipulated effects are excluded from the denominator when they are not the focal effect. For `aovlist` fits the names must match the effect labels visible in `summary(object)`.

SS_effect	Sum of squares for the focal effect (option 2).
SS_observed	Sums of squares for the measured factors. Scalar or numeric vector (option 2).
SS_error	Error (residual) sum of squares (option 2).
F_effect	Observed F-value for the focal effect (option 3).
df_effect	Numerator degrees of freedom for the focal effect (option 3).
F_observed	Vector of F-values for the measured factors (option 3).
df_observed	Numerator degrees of freedom for the measured factors, aligned with F_observed (option 3).
df_error	Error degrees of freedom (option 3).

Details

The function accepts *one* of three input interfaces:

1. a fitted model object (aov, lm, or aovlist for within-subjects / mixed designs) together with an observed vector listing which factors are measured (rather than manipulated);
2. raw sums of squares: SS_effect, SS_observed (one value per measured factor, or a scalar), and SS_error; or
3. raw F-values and degrees of freedom: F_effect, df_effect, F_observed (vector aligned with df_observed), and df_error.

If the user supplies both the SS interface (option 2) and the F/df interface (option 3), the function computes η^2_G from each and compares the results to within a 1e-6 tolerance. When the two interfaces agree, the SS value is returned. When they disagree, the function stops with a detailed message reporting both values.

Formula.

$$\hat{\eta}^2_G = \frac{SS_{\text{effect}}}{SS_{\text{effect}} + \sum_{\text{obs}} SS_{\text{measured}} + SS_{\text{error}}}.$$

For the F/df interface the equivalent ratio form is used, dividing through by SS_{error} so that no total-N argument is required: $SS_i / SS_{\text{error}} = F_i \cdot df_i / df_{\text{error}}$.

Designs supported.

- *Between-subjects ANOVA (single stratum).* The function reads anova(object). For each focal effect, the denominator is $SS_{\text{focal}} + \sum SS_{\text{measured (others)}} + SS_{\text{error}}$. Manipulated factors that are not the focal effect contribute nothing to the denominator.
- *Within-subjects and mixed ANOVA (*aovlist*, multi-stratum).* The function reads summary(object) and walks every error stratum. For each focal effect, the denominator is $SS_{\text{focal}} + \sum SS_{\text{measured (others)}} + \sum_s SS_{\text{error}(s)}$, where the last sum runs over every error stratum (both the between-subjects "subjects" stratum and any within-subjects error strata). This is the Olejnik & Algina (2003) / Bakeman (2005) rule that makes the subject-level variance act as an "always-measured" contributor in repeated measures designs.

Focal-effect self-exclusion. If the focal effect itself is listed in observed, the function excludes it from the observed-sum component (the focal effect's *own* SS already appears in the numerator and the leading term of the denominator). Practically this means listing every effect as observed reduces to total η^2 for between-subjects designs.

Higher-order interactions. An effect is a measured source of variance when *any* factor in its term is measured (Olejnik & Algina, 2003, Eq. 5; Bakeman, 2005), so listing a factor in `observed` also places every interaction containing that factor in the denominator automatically. In a design with manipulated `A` and measured `c`, `observed = "c"` therefore puts `c` and `A:c` in the denominator, which is what the cited papers' worked examples do. An explicit interaction label in `observed` is honored as given, declaring that one term measured without marking its constituent factors.

Covariates. Under Olejnik and Algina's Eq. 5, a covariate is a measured source whose SS always belongs in the denominator, so in an ANCOVA list the covariate in `observed`. Because `anova()` on an `lm`/`aov` fit uses sequential sums of squares, enter the covariate before the treatment factors in the model formula so its SS is adjusted the way the ANCOVA decomposition intends.

Confidence intervals. See `ci_eta_squared_generalized` for the corresponding CI function; both available CI methods are approximate and require independent evaluation.

Value

A `data.frame` with one row per focal effect. With a single-stratum fit and the raw interfaces the columns are `effect` and `eta_squared_generalized`; `effect` is `"overall"` for the raw interfaces. With an `aovlist` fit a `stratum` column is added, identifying which error stratum each effect came from.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bakeman, R. (2005). Recommended effect size statistics for repeated measures designs. *Behavior Research Methods, 37*(3), 379–384. doi:10.3758/BF03192707

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*, 137–152. doi:10.1037/a0028086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on η^2, Chapter 7 on factorial designs, and Chapter 11 on generalized η^2 for within-subjects designs.)

Olejnik, S., & Algina, J. (2003). Generalized eta and omega squared statistics: Measures of effect size for some common research designs. *Psychological Methods, 8*(4), 434–447. doi:10.1037/1082989X.8.4.434

See Also

`ci_eta_squared_generalized`, `eta_squared`, `eta_squared_partial`, `ci_omega_squared`

Other effect size estimates: `cles()`, `cliff_delta()`, `correction_for_attenuation()`, `eta_squared()`, `eta_squared_partial()`, `expected_partial_r()`, `expected_r()`, `expected_smd()`, `nnt_from_smd()`, `omega_squared()`, `omega_squared_partial()`, `probability_of_superiority_paired()`, `proportion_of_superiorit`
`responder_analysis()`, `smd_trimmed()`

Examples

```
# 1. Fitted model with `observed`. In the pygmalion expectancy
#        experiment, treatment is manipulated while grade is a measured
#        classification, so grade's variance stays in the denominator.
pyg <- pygmalion
pyg$grade <- factor(pyg$grade)
fit <- aov(iq_8 ~ treatment * grade, data = pyg)
eta_squared_generalized(fit, observed = "grade")

# 2. Raw sums of squares.
eta_squared_generalized(SS_effect = 100, SS_observed = c(40, 30),
                        SS_error = 200)

# 3. Raw F-values and degrees of freedom.
eta_squared_generalized(F_effect = 6.0, df_effect = 2,
                        F_observed = c(2.5, 1.8),
                        df_observed = c(1, 2), df_error = 50)

# 4. Within-subjects (repeated measures) ANOVA. The denominator
#        automatically includes the subject-level variance plus the
#        within-subjects error, following Bakeman (2005).
set.seed(113)
n <- 20
rm_data <- data.frame(
  subject = factor(rep(seq_len(n), each = 3)),
  time    = factor(rep(c("Pre", "Mid", "Post"), n),
                   levels = c("Pre", "Mid", "Post")),
  y       = rnorm(n, sd = 1.5)[rep(seq_len(n), each = 3)] +
            0.7 * rep(1:3, n) + rnorm(n * 3, sd = 1.2)
)
fit_rm <- aov(y ~ time + Error(subject/time), data = rm_data)
eta_squared_generalized(fit_rm)

# 5. Mixed design with a measured between-subjects factor. Treat
#        'group' as observed; its SS stays in the denominator for
#        'time' and the 'group:time' interaction.
set.seed(113)
n_per_group <- 10
n <- n_per_group * 2
mixed_data <- data.frame(
  subject = factor(rep(seq_len(n), each = 3)),
  group   = factor(rep(c("Treatment", "Control"), each = 3 * n_per_group)),
  time    = factor(rep(c("Pre", "Mid", "Post"), n),
                   levels = c("Pre", "Mid", "Post")),
  y       = rnorm(n, sd = 1)[rep(seq_len(n), each = 3)] +
            0.5 * rep(1:3, n) + rnorm(n * 3, sd = 1)
)
fit_mixed <- aov(y ~ group * time + Error(subject/time), data = mixed_data)
eta_squared_generalized(fit_mixed, observed = "group")
```

`eta_squared_partial` *Partial Eta Squared (Effect Size for ANOVA)*

Description

Computes the sample *partial* eta squared (η_p^2), the proportion of variance accounted for by a fixed effect after the variance attributable to the other effects in the model has been removed:

$$\hat{\eta}_p^2 = \frac{SS_{\text{effect}}}{SS_{\text{effect}} + SS_{\text{error}}} = \frac{df_{\text{effect}} \cdot F}{df_{\text{effect}} \cdot F + df_{\text{error}}}.$$

Accepts either the raw ANOVA summary (F, effect df, error df) or a fitted `aov`/`lm`/`aovlist` object, in which case the function returns one row per effect (with stratum identification for within-subjects fits).

Usage

```
eta_squared_partial(
  object = NULL,
  F_value = NULL,
  df_effect = NULL,
  df_error = NULL
)
```

Arguments

`object`	Optional. A fitted model object of class `aov`, `lm`, or `aovlist` (multi-stratum aov fit, e.g.\ `aov(y ~ A + Error(subject/A), data = d)`).
`F_value`	Observed F-value (ignored if `object` is supplied).
`df_effect`	Numerator degrees of freedom for the effect (ignored if `object` is supplied).
`df_error`	Error (residual) degrees of freedom (ignored if `object` is supplied).

Details

This function is the explicitly-named counterpart of `eta_squared`. The two share the same point-estimate formula, in a one-way ANOVA they coincide with *total* η^2; in a factorial or within-subjects ANOVA both functions return the per-effect *partial* value computed against that effect's own error stratum. `eta_squared_partial` is provided so that user code that explicitly intends partial η^2 carries that meaning in its name.

Designs supported. Single-stratum `aov`/`lm` fits and multi-stratum `aovlist` fits (within-subjects and mixed designs) are both handled by the model interface. For multi-stratum fits, each effect uses its own stratum's residual *df*, so a within-subjects factor's partial η^2 is computed against the within-subjects error and a between-subjects factor's is computed against the between-subjects error.

Value

A `data.frame` with one row per effect. Single-stratum fits and the raw interface return columns `effect`, `eta_squared_partial`, `F_value`, `df_effect`, `df_error`. `aovlist` (within-subjects / mixed) fits additionally include a `stratum` column identifying which error term each effect's F test came from. With the raw-argument interface `effect` is `"overall"`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1973). Eta-squared and partial eta-squared in fixed factor ANOVA designs. *Educational and Psychological Measurement, 33*(1), 107–112.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*, 137–152. doi:10.1037/a0028086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on η^2, Chapter 7 on factorial designs, and Chapter 11 on generalized η^2 for within-subjects designs.)

Steiger, J. H. (2004). Beyond the F test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

`ci_eta_squared_partial`, `eta_squared`

Other effect size estimates: `cles()`, `cliff_delta()`, `correction_for_attenuation()`, `eta_squared()`, `eta_squared_generalized()`, `expected_partial_r()`, `expected_r()`, `expected_smd()`, `nnt_from_smd()`, `omega_squared()`, `omega_squared_partial()`, `probability_of_superiority_paired()`, `proportion_of_superiorit`
`responder_analysis()`, `smd_trimmed()`

Examples

```
# Raw-argument interface.
eta_squared_partial(F_value = 11.221, df_effect = 4, df_error = 50)

# Factorial ANOVA: partial eta squared per effect (pygmalion data:
# expectancy treatment x grade, 2 x 6 with unequal cell sizes,
# N = 310). The treatment is manipulated; grade is a measured
# classification of the pupils.
fit <- aov(iq_8 ~ treatment * factor(grade), data = pygmalion)
eta_squared_partial(fit)

# Within-subjects ANOVA: per-effect partial eta squared with stratum.
set.seed(113)
n <- 20
```

```
rm_data <- data.frame(
  subject = factor(rep(seq_len(n), each = 3)),
  time    = factor(rep(c("Pre", "Mid", "Post"), n),
                   levels = c("Pre", "Mid", "Post")),
  y       = rnorm(n, sd = 1.5)[rep(seq_len(n), each = 3)] +
            0.7 * rep(1:3, n) + rnorm(n * 3, sd = 1.2)
)
fit_rm <- aov(y ~ time + Error(subject/time), data = rm_data)
eta_squared_partial(fit_rm)
```

`expected_partial_r` *Exact Expected Value of the Sample Partial Correlation*

Description

Computes $\mathrm{E}[r_{XY \cdot Z_1 \cdots Z_J} \mid \rho, n, J]$, the exact expected value of the sample partial Pearson correlation coefficient under multivariate normality, applying the Olkin-Pratt (1958) / Hotelling (1953) result for the simple Pearson correlation to the partial-correlation setting, an extension Olkin and Pratt (1958, Section 2.3) give themselves. Like the simple *r*, the partial *r* is downward-biased as an estimator of the population partial correlation $\rho_{XY \cdot Z}$, with the magnitude of the bias growing in the number of controls J and shrinking in the sample size n.

Usage

```
expected_partial_r(rho, n, J)
```

Arguments

`rho`	Population partial correlation $\rho_{XY \cdot Z_1 \cdots Z_J}$. Scalar or vector in $[-1, 1]$.
`n`	Total sample size; $n - J - 1 \geq 3$ is required for well-conditioned computation.
`J`	Number of variables partialled out (the count of $Z_1, \ldots, Z_J$); at least 1.

Details

The formula is the same as for the simple *r* but with the degrees of freedom reduced from n to $n - J$:

$$\mathrm{E}[r_{XY \cdot Z} \mid \rho_{XY \cdot Z}, n, J] \;=\; \rho_{XY \cdot Z} \cdot {}_2F_1\!\left(\tfrac{1}{2},\, \tfrac{1}{2};\, \tfrac{n-J+1}{2};\, \rho^2_{XY \cdot Z}\right) \cdot \frac{\Gamma\!\left(\frac{n-J}{2}\right)^2}{\Gamma\!\left(\frac{n-J-1}{2}\right)\,\Gamma\!\left(\frac{n-J+1}{2}\right)}.$$

`expected_partial_r()` is especially useful at the *design stage*: sample size plans that assume $\rho_{XY \cdot Z}$ as the value to be observed will under-deliver on the expected width or power of a CI on the partial correlation by an amount that grows in J.

Generalization of Olkin-Pratt. Under multivariate normality, the partial correlation $r_{XY \cdot Z}$ computed from a sample of size n has the same sampling distribution as a simple Pearson correlation from a sample of size $n - J$ (Anderson, 2003, Theorem 4.3.5, Section 4.3.2, p. 143, who credits the

derivation to Fisher, 1924). The bias formula for the simple *r* (Hotelling, 1953; Olkin & Pratt, 1958) therefore applies directly to the partial *r* with the substitution $n \to n - J$. The Olkin-Pratt (1958) unbiased estimator extends in the same way: given an observed $r_{XY \cdot Z}$, the unbiased estimator of $\rho_{XY \cdot Z}$ is $r \cdot {}_2F_1(1/2, 1/2; (n - J - 2)/2; 1 - r^2)$ (Olkin & Pratt, 1958, Section 2.3, where the partial correlation is shown to have the simple correlation's density at the reduced sample size, so their estimator applies with that substitution).

Magnitude of the bias. To leading order, from Hotelling's (1953) expansion of the simple-correlation expectation applied at the reduced sample size:

$$\rho_{XY \cdot Z} - \mathrm{E}[r_{XY \cdot Z}] \;\approx\; \rho_{XY \cdot Z}(1 - \rho^2_{XY \cdot Z})/[2(n - J - 1)],$$

matching the sign convention of the returned `bias` column, $\rho_{XY \cdot Z} - \mathrm{E}[r_{XY \cdot Z}]$, which is positive for positive $\rho_{XY \cdot Z}$. For $\rho_{XY \cdot Z} = 0.4$, $n = 30$, $J = 2$, the exact bias is about $+0.00624$; for $n = 30$, $J = 10$, about $+0.00888$.

Tuning the series convergence. The underlying ${}_2F_1$ series is summed by forward recurrence and stops when the relative contribution falls below `getOption("DMAR.expected_r.tol",1e-15)`, with a maximum of `getOption("DMAR.expected_r.max_iter", 5000L)` terms. These are the same options as for `expected_r`; users almost never need to change them.

Value

A `data.frame` with one row per (`rho`, `n`, `J`) input and the columns `rho`, `n`, `J`, `expected_partial_r`, `bias`, and `relative_bias`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Anderson, T. W. (2003). *An introduction to multivariate statistical analysis* (3rd ed.), Sections 4.2 and 4.3. Wiley. (Theorem 4.3.5, Section 4.3.2, p. 143: the sample partial correlation based on N observations, with J variables partialled out, has cdf $F[r \mid N - J, \rho]$; Anderson writes the count of partialled variables as $p - q$.)

Hotelling, H. (1953). New light on the correlation coefficient and its transforms. *Journal of the Royal Statistical Society, Series B, 15*(2), 193–232.

Olkin, I., & Pratt, J. W. (1958). Unbiased estimation of certain correlation coefficients. *The Annals of Mathematical Statistics, 29*(1), 201–211.

See Also

`expected_r`, `var_partial_r`, `ss_aipe_partial_r`

Other effect size estimates: `cles()`, `cliff_delta()`, `correction_for_attenuation()`, `eta_squared()`, `eta_squared_generalized()`, `eta_squared_partial()`, `expected_r()`, `expected_smd()`, `nnt_from_smd()`, `omega_squared()`, `omega_squared_partial()`, `probability_of_superiority_paired()`, `proportion_of_superiorit`
`responder_analysis()`, `smd_trimmed()`

Examples

```
# 1. A single value: rho = 0.4, n = 30, J = 3 controls.
expected_partial_r(rho = 0.4, n = 30, J = 3)

# 2. Bias grows with J at fixed (rho, n):
expected_partial_r(rho = 0.4, n = 30, J = c(1, 2, 5, 10, 20))

# 3. Partialing J = 1 variable at n = 11 is distributed exactly as a
#    simple correlation at n = 10 (Anderson, 2003, Theorem 4.3.5):
expected_partial_r(rho = 0.5, n = 11, J = 1)$expected_partial_r
expected_r(rho = 0.5, n = 10)$expected_r
```

---

expected_r *Exact Expected Value of the Sample Pearson Correlation Given ρ and n*

Description

Computes $\mathrm{E}[r \mid \rho, n]$, the exact expected value of the sample Pearson product-moment correlation coefficient under bivariate normality, using the Olkin-Pratt (1958) closed form built on Hotelling's (1953) exact density. The sample r is downward-biased as an estimator of the population ρ, a fact known since Soper (1913) and Fisher (1915); although this exact-bias equation has been available for more than half a century, it has historically not been easy to implement in general-purpose statistical software and is correspondingly rarely used in applied work. This function makes the exact bias visible so that users can decide whether to apply the Olkin-Pratt (1958) unbiased estimator of ρ (see Details).

Usage

```
expected_r(rho, n)
```

Arguments

rho	Population correlation coefficient. A numeric scalar or vector in the interval $[-1, 1]$.
n	Sample size on which the Pearson r would be computed. A scalar or vector of integers with $n \geq 4$. (At $n = 3$ the bias is defined but the exact formula is numerically delicate; we require $n \geq 4$ for well-conditioned computation.) When rho and n are both vectors they must have the same length or recycle cleanly.

Details

expected_r() is especially useful at the *design stage* of a study, where sample size planning typically proceeds from an assumed value of the population correlation ρ. Because the value of r a researcher should expect to observe on average is *smaller in absolute value* than the assumed ρ,

plugging ρ directly into sampling-distribution machinery (precision of r, width of a confidence interval, power of a test of $H_0 : \rho = 0$) systematically over-promises on the realized precision or power. Substituting `expected_r(rho, n)` for the bare ρ in such planning calculations corrects the leading-order over-promise. See **Examples** for a design-stage walk-through.

The exact formula. Under bivariate normality, if r is the sample Pearson correlation in a sample of size n,

$$\mathrm{E}[r \mid \rho, n] \;=\; \rho \cdot {}_2F_1\!\left(\tfrac{1}{2},\, \tfrac{1}{2};\, \tfrac{n+1}{2};\, \rho^2\right) \cdot \frac{\Gamma\!\left(\tfrac{n}{2}\right)^2}{\Gamma\!\left(\tfrac{n-1}{2}\right)\,\Gamma\!\left(\tfrac{n+1}{2}\right)}.$$

Here ${}_2F_1(a, b; c; z) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k\, k!}\, z^k$ is the Gauss hypergeometric function with Pochhammer symbols $(x)_k = x(x+1)\cdots(x+k-1)$ (Hotelling, 1953, Section 7 for the moments of r, Section 3, equation 25, for the exact density; Olkin & Pratt, 1958, equation 3.2, for this closed form). For $\rho^2 < 1$ the series converges absolutely; this implementation sums by a numerically stable forward recurrence and stops when the relative contribution of the next term falls below a tolerance.

Tuning the series convergence (rarely needed). Two internal tuning constants control the ${}_2F_1$ series summation:

- `DMAR.expected_r.tol`, relative tolerance for stopping the series (default `1e-15`).
- `DMAR.expected_r.max_iter`, maximum number of series terms before issuing a non-convergence warning (default `5000L`).

Both have sensible defaults; advanced users who need different values (e.g., for very near-boundary $|\rho|$ where the series converges slowly) can set them via `options()`, for example `options(DMAR.expected_r.tol = 1e-12)`. They are deliberately hidden from the function signature so as not to clutter the everyday user's view of the call.

Why the bias is present even though s^2 is unbiased for σ^2. The downward bias arises because r is a nonlinear function of unbiased sample moments. By Jensen's inequality and the concavity of the square root in the denominator of r, the expectation of the ratio is not the ratio of the expectations. The bias is largest when n is small or $|\rho|$ is moderate; as $n \to \infty$, $\mathrm{E}[r] \to \rho$.

Sign and magnitude. The bias $\rho - \mathrm{E}[r]$ has the same sign as ρ and is approximately $\rho(1 - \rho^2)/[2(n-1)]$ to leading order (Fisher, 1915; Hotelling, 1953; Ghosh, 1966); the exact formula above incorporates all higher-order corrections. For $\rho = 0.5$, $n = 10$, the bias is about $+0.021$; for $\rho = 0.5$, $n = 30$, about $+0.0065$.

Olkin-Pratt (1958) unbiased estimator of ρ. The companion to this expected-value calculation is the Olkin-Pratt unbiased estimator of ρ given an observed r:

$$\tilde{\rho}_{\mathrm{OP}}(r, n) \;=\; r \cdot {}_2F_1\!\left(\tfrac{1}{2},\, \tfrac{1}{2};\, \tfrac{n-2}{2};\, 1 - r^2\right).$$

Olkin & Pratt (1958) prove $\mathrm{E}[\tilde{\rho}_{\mathrm{OP}}(r, n) \mid \rho, n] = \rho$ exactly, for every $\rho \in (-1, 1)$ and $n \geq 4$. The commonly quoted first-order approximation $\tilde{\rho} \approx r\,[1 + (1 - r^2)/(2(n-3))]$ (Olkin, 1967) is the truncation of the OP series at $k = 1$; the full series is what makes the estimator exactly unbiased.

Although the unbiased estimator is straightforward to apply, it is rarely used because in most downstream uses (significance testing, Fisher's Z confidence intervals, structural-equation models) the bias is small relative to other sources of uncertainty. Where it does matter, meta-analyses with many small samples, reliability / validity coefficients estimated from short calibration samples, design-stage estimates feeding into AIPE sample size machinery the correction is well worth applying.

Connection to Fisher's Z transform. The variance-stabilizing transform $z = \tanh^{-1}(r)$, proposed in passing in Fisher (1915, p. 521) and developed in Fisher (1921), has approximate variance

$1/(n-3)$ regardless of ρ, but $\mathrm{E}[z]$ also carries a small-sample bias of order $1/n$ (Hotelling, 1953, Section 8). Hotelling (1953, Sections 9–10) gives bias-adjusted and variance-stabilized refinements of Z, e.g. $z - (3z + r)/(4n)$.

Value

A `data.frame` with one row per (`rho`, `n`) input and the columns

- `rho`, the input population correlation,
- `n`, the input sample size,
- `expected_r`, $\mathrm{E}[r \mid \rho, n]$,
- `bias`, $\rho - \mathrm{E}[r \mid \rho, n]$ (the amount by which the sample r underestimates ρ on average),
- `relative_bias`, `bias / rho` when $\rho \neq 0$, and `NA` when $\rho = 0$.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Anderson, T. W. (2003). *An introduction to multivariate statistical analysis* (3rd ed.), Section 4.2. Wiley.

Fisher, R. A. (1915). Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika, 10*(4), 507–521.

Fisher, R. A. (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron, 1*, 3–32.

Ghosh, B. K. (1966). Asymptotic expansions for the moments of the distribution of correlation coefficient. *Biometrika, 53*(1/2), 258–262.

Hotelling, H. (1953). New light on the correlation coefficient and its transforms. *Journal of the Royal Statistical Society, Series B, 15*(2), 193–232. (Discussion, pp.\ 225–232.)

Olkin, I. (1967). Correlations revisited. In J. C. Stanley (Ed.), *Improving experimental design and statistical analysis* (pp.\ 102–128). Rand McNally.

Olkin, I., & Pratt, J. W. (1958). Unbiased estimation of certain correlation coefficients. *The Annals of Mathematical Statistics, 29*(1), 201–211.

Soper, H. E. (1913). On the probable error of the correlation coefficient to a second approximation. *Biometrika, 9*(1/2), 91–115.

Soper, H. E., Young, A. W., Cave, B. M., Lee, A., & Pearson, K. (1917). On the distribution of the correlation coefficient in small samples. Appendix II to the papers of "Student" and R. A. Fisher: A cooperative study. *Biometrika, 11*(4), 328–413.

Stuart, A., & Ord, J. K. (1994). *Kendall's advanced theory of statistics, Vol.\ 1: Distribution theory* (6th ed.), Section 16.32. Edward Arnold.

See Also

ci_r, cor, cor.test

Other effect size estimates: cles(), cliff_delta(), correction_for_attenuation(), eta_squared(), eta_squared_generalized(), eta_squared_partial(), expected_partial_r(), expected_smd(), nnt_from_smd(), omega_squared(), omega_squared_partial(), probability_of_superiority_paired(), proportion_of_superiority(), responder_analysis(), smd_trimmed()

Examples

```
# 1. A single value: rho = 0.5, n = 10. The sample r is downwardly
#        biased by about 0.021, roughly 4% of rho.
expected_r(rho = 0.5, n = 10)

# 2. Bias as a function of n for fixed rho. The bias is roughly
#        rho * (1 - rho^2) / (2(n - 1)) to leading order; as n grows it
#        shrinks toward zero.
expected_r(rho = 0.5, n = c(5, 10, 20, 50, 100, 500))

# 3. Bias as a function of rho for fixed n. The bias is zero at
#        rho = 0 and rho = +/- 1, and largest near rho = +/- 0.6.
expected_r(rho = seq(0, 0.95, by = 0.05), n = 10)

# 4. The Olkin-Pratt unbiased estimator: invert the bias for an
#        observed sample r.
set.seed(113)
x <- rnorm(20); y <- 0.4 * x + sqrt(1 - 0.4^2) * rnorm(20)
r_obs <- cor(x, y)
r_obs

# Olkin-Pratt unbiased estimator of rho:
op_unbiased <- function(r, n, tol = 1e-15, max_iter = 5000) {
  z <- 1 - r^2; c_par <- (n - 2) / 2
  s <- 1; term <- 1
  for (k in seq_len(max_iter)) {
    term <- term * ((k - 0.5)^2) / ((c_par + k - 1) * k) * z
    s <- s + term
    if (abs(term) < tol * abs(s)) break
  }
  r * s
}
op_unbiased(r_obs, n = 20)

# 5. Design-stage use: a study planned around rho = 0.4 with n = 30.
#        Naive plug-in says we expect to observe r = 0.40 on average,
#        but the realized expected r is smaller, and that gap matters
#        for any precision- or power-based sample size calculation that
#        plugs in rho as if it were the expected sample r.
rho_planned <- 0.4
n_planned   <- 30
expected_r(rho = rho_planned, n = n_planned)
```

```
# 6. Tuning constants are hidden from the signature but tunable
#        through options() for the rare cases that need them (e.g.,
#        very near-boundary |rho| where the 2F1 series converges
#        slowly). The defaults rarely need to be changed.
options(DMAR.expected_r.tol = 1e-12,
        DMAR.expected_r.max_iter = 20000L)
expected_r(rho = 0.999, n = 5)
options(DMAR.expected_r.tol = NULL,
        DMAR.expected_r.max_iter = NULL)   # restore defaults
```

`expected_R2` *Expected Value of the Squared Multiple Correlation Coefficient*

Description

Computes the expected value of the observed squared multiple correlation coefficient given the population squared multiple correlation coefficient, the sample size, and the number of predictors. The sample R^2 is a positively biased estimator of its population value, and the expected value quantifies how large that bias is for a particular design.

Usage

```
expected_R2(population_R2, N, p)
```

Arguments

`population_R2`	Population squared multiple correlation coefficient
`N`	Sample size
`p`	The number of predictor variables

Details

Uses the hypergeometric function as discussed in section 28 of Stuart, Ord, and Arnold (1999) in order to obtain the *correct* value for the squared multiple correlation coefficient. Many times an exact value is given that ignores the hypergeometric function. This function yields the correct value.

Value

A 1-row `data.frame` with columns `term` and `value`. The `term` value is `"expected_value_population_R2"` and `value` is the expected value of R^2 under random sampling.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*, 524–555. doi:10.1080/00273170802490632

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on R^2 as a model comparison effect size.)

Olkin, I., & Pratt, J. W. (1958). Unbiased estimation of certain correlation coefficients. *The Annals of Mathematical Statistics, 29*(1), 201–211.

Stuart, A., Ord, J. K., & Arnold, S. (1999). *Kendall's advanced theory of statistics, volume 2A: Classical inference and the linear model* (6th ed.). Arnold.

See Also

ss_aipe_R2, ci_R2, var_R2

Examples

```
expected_R2(.5, 10, 5)
expected_R2(.5, 25, 5)
expected_R2(.5, 50, 5)
expected_R2(.5, 100, 5)
expected_R2(.5, 1000, 5)
expected_R2(.5, 10000, 5)
```

---

expected_smd *Exact Expected Value of Cohen's* d *(and Hedges'* g *Bias Correction)*

Description

Computes $\mathrm{E}[\hat{d} \mid \delta, n_1, n_2]$, the exact expected value of the sample standardized mean difference (Cohen's *d*, with pooled variance) under bivariate normality and the noncentral *t* sampling distribution (Hedges, 1981). The sample d is upward-biased as an estimator of the population δ: $\mathrm{E}[\hat{d}] = \delta / J(df)$, where

$$J(df) = \frac{\Gamma(df/2)}{\sqrt{df/2}\,\Gamma((df-1)/2)}$$

is Hedges' (1981) bias-correction factor (< 1 for finite df, tending to 1 as $n \to \infty$). Hedges' g, the unbiased estimator of δ, is then $g = J \cdot \hat{d}$. The same $J(df)$ is the workhorse of smd when unbiased = TRUE.

Usage

```
expected_smd(delta, n_1, n_2 = NULL)
```

Arguments

delta	Population standardized mean difference. A numeric scalar or vector.
n_1	Sample size in the first group. Scalar or vector.
n_2	Sample size in the second group. Scalar or vector. If omitted, defaults to n_1 (balanced design).

Details

expected_smd() is especially useful at the *design stage* of a study, where sample size planning typically proceeds from an assumed population standardized mean difference δ. Because the value of $\hat{d}$ a researcher should expect to observe on average is *larger in absolute value* than δ, plugging δ directly into sampling-distribution machinery (precision of $\hat{d}$, width of a confidence interval, power of a test of $H_0 : \delta = 0$) over-promises on the realized precision when the precision is expressed on the $\hat{d}$ scale. Substituting expected_smd(delta, n_1, n_2) for the bare δ corrects this leading-order bias.

Derivation. Under bivariate normality with equal variances, the observed *t*-statistic $t = \hat{d}\sqrt{n_1 n_2/(n_1+n_2)}$ follows a noncentral *t* distribution with $df = n_1 + n_2 - 2$ degrees of freedom and noncentrality parameter $\lambda = \delta\sqrt{n_1 n_2/(n_1+n_2)}$. The expected value of a noncentral *t* variate equals $\lambda/J(df)$ (Johnson, Kotz, & Balakrishnan, 1995, Section 31.3), so $\mathrm{E}[\hat{d}] = \mathrm{E}[t]/\sqrt{n_1 n_2/(n_1+n_2)} = \delta/J(df)$. The bias $\mathrm{E}[\hat{d}] - \delta = \delta\,(1-J)/J$ is positive when $\delta > 0$.

Magnitude of the correction. $J(df) \approx 1 - 3/(4\,df - 1)$ to leading order (Hedges & Olkin, 1985, p.\ 81). For $\delta = 0.5$, $n_1 = n_2 = 10$ ($df = 18$), $J \approx 0.957$, so $\mathrm{E}[\hat{d}] \approx 0.522$ and the upward bias is about 4%. For $n_1 = n_2 = 50$ the bias is under 1%; for $n_1 = n_2 = 5$ (very small samples) it exceeds 10%.

Connection to Hedges' *g*. The natural inverse of this function is Hedges' g: given an observed $\hat{d}$, the unbiased estimator of δ is $g = J(df)\hat{d}$, which satisfies $\mathrm{E}[g \mid \delta] = \delta$ exactly under the same noncentral *t* model. smd with unbiased = TRUE returns g.

Value

A data.frame with one row per (delta, n_1, n_2) input and the columns

- delta, the population SMD,
- n_1, n_2, group sample sizes,
- expected_smd, $\mathrm{E}[\hat{d} \mid \delta]$,
- bias, $\mathrm{E}[\hat{d}] - \delta$ (the amount by which $\hat{d}$ overestimates δ on average),
- j_correction, the Hedges (1981) correction factor $J(df)$ used to compute the unbiased g, where $df = n_1 + n_2 - 2$.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Hedges, L. V. (1981). Distribution theory for Glass's estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. Academic Press. (See Section 5, equations 6 and 9.)

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous univariate distributions, volume 2* (2nd ed.), Section 31.3. Wiley.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

See Also

smd, ci_smd, ss_aipe_smd, expected_r, expected_R2

Other effect size estimates: cles(), cliff_delta(), correction_for_attenuation(), eta_squared(), eta_squared_generalized(), eta_squared_partial(), expected_partial_r(), expected_r(), nnt_from_smd(), omega_squared(), omega_squared_partial(), probability_of_superiority_paired(), proportion_of_superiority(), responder_analysis(), smd_trimmed()

Examples

```
# 1. Balanced design: delta = 0.5, n = 10 per group.
expected_smd(delta = 0.5, n_1 = 10)

# 2. Bias as a function of n at fixed delta.
expected_smd(delta = 0.5, n_1 = c(5, 10, 20, 50, 100, 500))

# 3. Unbalanced design.
expected_smd(delta = 0.5, n_1 = 30, n_2 = 60)

# 4. Design-stage use: an a-priori delta of 0.4, planned n of 40/group.
#        The d we should expect to observe on average is slightly larger.
expected_smd(delta = 0.4, n_1 = 40)
```

---

factorial_anova *Between-Subjects Factorial ANOVA for Unbalanced Designs*

Description

Fits a between-subjects factorial ANOVA (two or more crossed factors) and returns each effect's sum of squares, F test, and partial η^2 and partial ω^2 with confidence intervals, using a sum-of-squares type the user chooses. It is built for the case that makes the choice matter, an *unbalanced* design (unequal cell sizes), where the Type I, II, and III sums of squares differ. For a balanced design the three types coincide and the choice is immaterial.

Usage

```
factorial_anova(formula, data, ss_type = 3L, conf_level = 0.95)
```

Arguments

formula	A two-sided formula naming the numeric response and the crossed factors, for example y ~ A * B or y ~ A * B * C. Predictors are coerced to factors. Write the full crossing you want tested; A * B expands to A + B + A:B.
data	A data.frame containing the response and the factors.
ss_type	The sum-of-squares type: 1, 2, or 3 (equivalently "I", "II", or "III"). Default 3 (Type III), the common reporting default. See Details for what each type tests.
conf_level	Confidence level for the effect size confidence intervals. Default 0.95.

Details

Every sum of squares is computed as a model comparison, the increase in error sum of squares when an effect's parameters are removed from a model, following the model comparison development of Maxwell, Delaney, and Kelley (2027, Chapter 7). The computation uses only base R (stats); it does not depend on the **car** package, though it agrees with car::Anova() to numerical precision.

The types as model comparisons. A sum of squares for an effect is the increase in the error sum of squares when the effect's parameters are dropped from the model, SS = E(restricted) - E(full). The three conventional types differ only in which other effects the full and restricted models hold in common (Maxwell, Delaney, and Kelley, 2027, Chapter 7; Overall and Spiegel, 1969):

- **Type I (sequential).** Each effect is adjusted only for the effects listed before it in formula: SS(A), then SS(B | A), then SS(A:B | A, B). The parts sum to the model sum of squares, but the answer depends on the order of the terms.
- **Type II.** Each effect is adjusted for every other effect that does not contain it (it respects marginality): a main effect is adjusted for the other main effects but not for the interactions that contain it. Type II is the most powerful choice when the interaction is null and does not depend on how the factors are coded (Overall and Spiegel's Method 2; Appelbaum and Cramer, 1974).
- **Type III.** Each effect is adjusted for all other effects, including the higher order interactions that contain it. This tests each main effect as a contrast on the unweighted marginal means, so it is computed here with sum-to-zero contrasts (contr.sum), which is what makes the Type III main-effect test the intended one. It is the default reported by many programs.

For a balanced design the effects are orthogonal and the three types are identical; the distinction is a property of unbalanced (nonorthogonal) data.

Reading the main effects when an interaction is present. When an interaction is real, the marginal main-effect tests, of any type, are usually not the question of interest; examine the interaction and the simple effects instead (Maxwell, Delaney, and Kelley, 2027). See the vignette *Sums of Squares in Nonorthogonal Designs* for a worked comparison.

Effect sizes. Partial η^2 and partial ω^2 are formed for each effect from its F, its degrees of freedom, and the error degrees of freedom, with noncentral F confidence intervals (ci_eta_squared_partial, ci_omega_squared). The confidence limits are those of the interval for the population proportion of variance the effect accounts for (Kelley, 2007). Partial η^2 and partial ω^2 are two point estimators

of that same population quantity, partial ω^2 correcting the upward bias of partial η^2, so the two estimators differ but share the interval.

Estimability. All cells must be filled. If a factor combination is empty the design is rank deficient and the factorial effects are not all estimable; the function stops with a message rather than return a value that depends on an arbitrary choice.

Value

A `data.frame` (class `dmar_tbl`) with one row per effect plus a `Residuals` row. Columns are the effect label (`effect`), the sum of squares (`SS`), degrees of freedom (`df`), the F statistic (`F_value`) and its p-value (`p_value`), and partial η^2 and partial ω^2 with their lower and upper confidence limits. The chosen sum-of-squares type is recorded on the object (`attr(x, "ss_type")`) and printed beneath the table. The residual row carries only `SS` and `df`. Stored values keep full precision; the display rounds (see `dmar_tbl`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Appelbaum, M. I., & Cramer, E. M. (1974). Some problems in the nonorthogonal analysis of variance. *Psychological Bulletin, 81*(6), 335–343.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 7 on higher order between-subjects designs.)

Overall, J. E., & Spiegel, D. K. (1969). Concerning least squares analysis of experimental data. *Psychological Bulletin, 72*(5), 311–322.

See Also

`ancova` for a covariate-adjusted one-way design, `mixed_anova` for mixed-model (fixed and random) F ratios, `manova_split_plot` for the multivariate mixed design, and `ci_eta_squared_partial` / `ci_omega_squared` for the effect size intervals.

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `correlations_test()`, `equivalence_r()`, `equivalence_smd()`, `manova_split_plot()`, `mauchly_test()`, `mixed_anova()`, `obrien_test()`, `pairwise_within()`, `randomization_test()`, `randomization_test_paired()`, `regions_of_significance()`, `simple_effects_AB()`, `summary_t_test()`, `welch_t()`

Examples

```
# An unbalanced two-factor design: IQ gain in the pygmalion expectancy
# experiment, by treatment and by lower (grades 1 and 2) versus upper
# (grades 3 through 6) grades, where the expectancy effect concentrated
# in the lower grades. The cell sizes are unequal, so the Types differ.
```

```
pyg <- pygmalion
pyg$grade_band <- factor(ifelse(pyg$grade <= 2, "lower", "upper"))
factorial_anova(iq_gain ~ treatment * grade_band, data = pyg)

# The same design under Type II (adjusts each main effect for the other
# main effect, but not for the interaction). Here the main effect F
# statistics drop, and a warning reports that the affected noncentral F
# lower limits are clamped to 0.
factorial_anova(iq_gain ~ treatment * grade_band, data = pyg, ss_type = 2)
```

`fleiss_kappa` *Fleiss's Kappa for Inter-Rater Agreement Among Multiple Raters*

Description

Computes Fleiss's (1971) kappa coefficient of agreement among $m \geq 2$ raters who classify each of N subjects into one of k nominal categories. Returns the point estimate together with the asymptotic standard error and the Wald confidence interval, plus a test of $H_0 : \kappa = 0$.

Usage

```
fleiss_kappa(
  ratings,
  conf_level = 0.95,
  ci_method = c("wald", "percentile", "bca"),
  B = 10000L,
  seed = NULL
)
```

Arguments

`ratings`	A numeric $N \times k$ matrix or `data.frame`: row i gives the counts of raters who assigned each of the k categories to subject i. Each row must sum to the same value m (the common number of raters per subject).
`conf_level`	Confidence level for the interval (default `0.95`).
`ci_method`	Interval method: `"wald"` (the default, the asymptotic interval from the Gwet (2008) linearization variance), `"percentile"` (bootstrap percentile), or `"bca"` (bootstrap bias-corrected and accelerated). The multirater kappa variance literature is unsettled, and Zapf, Castell, Morawietz, and Karch (2016) recommend bootstrap intervals in this setting: the subjects (rows) are resampled with replacement `B` times, kappa is recomputed on each resample, and the interval is read off the bootstrap distribution; the BCa variant additionally adjusts the quantile positions for median bias and acceleration (Efron & Tibshirani, 1993). The `se`, `z_value`, and `p_value` columns keep their asymptotic definitions under every `ci_method`; only the interval changes.

B	Number of bootstrap replications when ci_method is "percentile" or "bca" (default 10000; ignored for "wald").
seed	Optional integer seed for the bootstrap. The default NULL uses the current state of the random number generator; a supplied seed is set internally and the prior state restored on exit.

Details

For n_{ij} = the number of raters who assigned subject i to category j, with $\sum_j n_{ij} = m$ for every i, define the marginal proportion of category j as $p_j = \sum_i n_{ij}/(Nm)$, and the per-subject agreement

$$P_i = \frac{1}{m(m-1)}\Big(\sum_j n_{ij}^2 - m\Big).$$

Then Fleiss's kappa is

$$\hat\kappa_F = \frac{\bar P - P_e}{1 - P_e}, \qquad \bar P = \frac{1}{N}\sum_i P_i, \qquad P_e = \sum_j p_j^2.$$

Standard error. Two variances are involved, because the variance of $\hat\kappa_F$ under $H_0 : \kappa = 0$ is not its variance at a nonzero value. The test of no agreement uses the null variance of Fleiss, Nee, and Landis (1979, Equation 12), who corrected the standard errors given in Fleiss (1971),

$$\mathrm{Var}_0(\hat\kappa_F) = \frac{2\left(P_e + P_e^2 - 2\sum_j p_j^3\right)}{N\,m\,(m-1)\,(1-P_e)^2},$$

and the reported z statistic and p-value come from it. On the Fleiss (1971) Table 1 example below this gives $z = 17.65$, matching `irr::kappam.fleiss`. The confidence interval instead uses the linearization variance of Gwet (2008, Section 6), which is consistent at the estimated $\hat\kappa_F$: each subject i contributes an influence value κ_i^* (Gwet's Equations 34 and 35), and $\mathrm{Var}(\hat\kappa_F) = \sum_i(\kappa_i^* - \hat\kappa_F)^2/\{N(N-1)\}$, Gwet's Equation 33 with the sampling fraction set to zero. (Gwet derives the variance for the multiple-rater pi statistic, which is the same estimator as Fleiss's kappa.) The Wald confidence interval is $\hat\kappa_F \pm z_{1-\alpha/2}\,\widehat{\mathrm{SE}}$, with the upper limit truncated at 1. Using the null variance for the interval would understate the standard error and give a spuriously narrow interval.

Bootstrap interval. The variance of multirater kappa is unsettled in the literature, and Zapf, Castell, Morawietz, and Karch (2016) recommend a bootstrap interval in this setting. With `ci_method` = `"percentile"` or `"bca"` the subjects (the rows of `ratings`) are resampled with replacement `B` times, kappa is recomputed on each resample, and the interval is read off the bootstrap distribution: the percentile interval takes the empirical quantiles, and the BCa interval adjusts the quantile positions for median bias and for acceleration (Efron & Tibshirani, 1993). Ask for it when N is small or $\hat\kappa_F$ is near a boundary, where the Wald interval's coverage is least dependable. The `se`, `z_value`, and `p_value` columns keep their asymptotic definitions under every `ci_method`; only the interval changes. Bootstrap results vary from run to run; supply `seed` for reproducibility.

Fleiss's kappa is purely nominal (no weighting). For ordinal categories with two raters, use cohen_kappa with quadratic weights; for ordinal categories with three or more raters, an extension based on the intraclass correlation (icc) is more appropriate.

Value

A one-row `data.frame` (class `dmar_tbl`) with columns `kappa`, `se` (asymptotic standard error of $\hat{\kappa}_F$ used for the interval), `lower_limit`, `upper_limit`, `z_value`, `p_value` (Wald test of $H_0 : \kappa = 0$), `n_subjects`, `n_raters` (m), and `n_categories` (k).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin, 76*(5), 378–382.

Fleiss, J. L., Nee, J. C. M., & Landis, J. R. (1979). Large sample variance of kappa in the case of different sets of raters. *Psychological Bulletin, 86*(5), 974–977.

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, NY: Chapman & Hall/CRC.

Gwet, K. L. (2008). Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology, 61*(1), 29–48. doi:10.1348/000711006X126600

Zapf, A., Castell, S., Morawietz, L., & Karch, A. (2016). Measuring inter-rater reliability for nominal data: Which coefficients and confidence intervals are appropriate? *BMC Medical Research Methodology, 16*, 93. doi:10.1186/s1287401602009

See Also

`cohen_kappa`, `icc`

Other reliability: `cohen_kappa()`, `diagnosis_agreement`, `icc()`, `reliability()`, `reliability_H()`, `reliability_alpha()`, `reliability_kr20()`, `reliability_omega()`, `reliability_omega_categorical()`

Examples

```
# Fleiss (1971) Table 1 example: 30 subjects rated by 6 raters into
# 5 diagnostic categories (Depression, Personality Disorder,
# Schizophrenia, Neurosis, Other). Each row of `ratings` gives, for one
# subject, the count of raters who chose each category (rows sum to 6).
# kappa = 0.430, matching Fleiss (1971).
fleiss_1971 <- matrix(c(
  0, 0, 0, 6, 0,
  0, 3, 0, 0, 3,
  0, 1, 4, 0, 1,
  0, 0, 0, 0, 6,
  0, 3, 0, 3, 0,
  2, 0, 4, 0, 0,
  0, 0, 4, 0, 2,
  2, 0, 3, 1, 0,
  2, 0, 0, 4, 0,
  0, 0, 0, 0, 6,
  1, 0, 0, 5, 0,
```

```
  1, 1, 0, 4, 0,
  0, 3, 3, 0, 0,
  1, 0, 0, 5, 0,
  0, 2, 0, 3, 1,
  0, 0, 5, 0, 1,
  3, 0, 0, 1, 2,
  5, 1, 0, 0, 0,
  0, 2, 0, 4, 0,
  1, 0, 2, 0, 3,
  0, 0, 0, 0, 6,
  0, 1, 0, 5, 0,
  0, 2, 0, 1, 3,
  2, 0, 0, 4, 0,
  1, 0, 0, 4, 1,
  0, 5, 0, 1, 0,
  4, 0, 0, 0, 2,
  0, 2, 0, 4, 0,
  1, 0, 5, 0, 0,
  0, 0, 0, 0, 6
), nrow = 30, byrow = TRUE)
fleiss_kappa(fleiss_1971)

# A percentile bootstrap interval, which resamples the subjects (rows)
# with replacement and recomputes kappa on each resample. Compare its
# limits with the Wald interval above; z_value and p_value keep their
# asymptotic definitions. B = 2000 keeps the example quick; a reported
# interval deserves the default B = 10000.
fleiss_kappa(fleiss_1971, ci_method = "percentile", B = 2000,
             seed = 113)
```

`format_p` *Format* p*-values for Display the DMAR Way*

Description

Render a vector of *p*-values as character strings at a fixed number of decimal places (default 4) with a floor label "< 10^(-digits_p)" for values too small to express. Never uses scientific notation. This is the package-wide convention for displaying *p*-values used by the `dmar_tbl` print layer, by `correlations_test`, and by the helper display functions `print_anova` and `print_summary`. Exposing it as a user-facing function lets analysts apply the same convention to ad hoc *p*-values that they want to report in prose or in a manually constructed table.

Usage

```
format_p(p, digits_p = 4L)
```

Arguments

p	Numeric vector of *p*-values. NA values pass through as NA_character_.
digits_p	Integer number of decimal places. Default 4L. Values strictly below $10^{-\text{digits\_p}}$ render as the floor label "< 10^(-digits_p)" (for example "< 0.0001" when digits_p = 4).

Details

The function does not modify the input value; the underlying numeric *p*-value retains full precision and can still be indexed out of whatever object holds it. Only the returned display string is rounded.

Value

A character vector of the same length as p.

Author(s)

Ken Kelley

See Also

print_anova, print_summary, dmar_tbl.

Examples

```
# Round to four decimals with a "< 0.0001" floor:
format_p(c(0.5, 0.0234, 0.0001234, 1e-10, NA))

# Six decimals when more precision is wanted:
format_p(0.0001234, digits_p = 6)

# Use inline in prose for a publication-style summary:
fit <- lm(weight ~ Time + Diet, data = ChickWeight)
p_time <- summary(fit)$coefficients["Time", "Pr(>|t|)"]
paste0("The Time coefficient was significant (p = ", format_p(p_time), ").")
```

---

glance.dmar_cfa_k *Glance at a Multiple-Factor CFA Fit*

Description

Returns a one-row data.frame of model-level summaries from a cfa_k table, in the column convention used by the **broom** ecosystem.

Usage

```
## S3 method for class 'dmar_cfa_k'
glance(x, ...)
```

Arguments

x	A dmar_cfa_k object returned by cfa_k.
...	Unused.

Value

A one-row data.frame with columns chi_square, df, p_value, cfi, tli, rmsea, rmsea_low, rmsea_high, srmr, AIC, BIC, logLik.

Author(s)

Ken Kelley <kkelley@nd.edu>

---

glance.dmar_mediation_mbco

Glance at an MBCO Mediation Fit

Description

Returns a one-row data.frame of full-model summaries from a mediation_mbco table, in the column convention used by the **broom** ecosystem.

Usage

```
## S3 method for class 'dmar_mediation_mbco'
glance(x, ...)
```

Arguments

x	A dmar_mediation_mbco object returned by mediation_mbco.
...	Unused.

Value

A one-row data.frame with columns nobs, npar, deviance, AIC, BIC, logLik.

Author(s)

Ken Kelley <kkelley@nd.edu>

glance.dmar_reliability *Glance at a Reliability Coefficient Estimate*

Description

Returns a one-row `data.frame` of model-level summaries in the column convention used by the **broom** ecosystem (`coefficient`, `estimate`, `se`, `ci_lower`, `ci_upper`, `conf_level`, `nobs`, `n_items`, `ci_method`).

Usage

```
## S3 method for class 'dmar_reliability'
glance(x, ...)
```

Arguments

`x`	A `dmar_reliability` object.
`...`	Unused.

Value

A one-row `data.frame`.

Author(s)

Ken Kelley <kkelley@nd.edu>

glance.mlmr *Glance at an Mlmr Fit*

Description

Returns a one-row `data.frame` of model-level summaries in the column convention used by the **broom** ecosystem (`R2`, `adj_R2`, `sigma`, `statistic`, `p_value`, `df`, `logLik`, `AIC`, `BIC`, `deviance`, `df_residual`, `nobs`). The `statistic` and `p_value` columns report the omnibus likelihood ratio test of all slopes equal to zero (the FIML analog of the `lm` omnibus *F*-test).

Usage

```
## S3 method for class 'mlmr'
glance(x, ...)
```

Arguments

x	An object of class "mlmr".
...	Unused.

Value

A one-row data.frame.

Author(s)

Ken Kelley <kkelley@nd.edu>

Examples

```
fit <- mlmr(t6_paragraph_comprehension ~ t5_general_information +
              t9_word_meaning,
            data = holzinger_swineford, ci_method = "wald")
generics::glance(fit)
```

---

gwet_ac *Gwet's AC1 and AC2 Chance-Corrected Agreement Coefficients*

Description

Computes Gwet's AC1 (nominal data; Gwet, 2008) and AC2 (ordinal data with user-supplied weights; Gwet, 2014) chance-corrected agreement coefficients for two or more raters. AC1/AC2 are more robust than Cohen's κ to extreme marginal-prevalence imbalance and the trait-distribution paradox.

Usage

```
gwet_ac(
  ratings,
  weights = c("unweighted", "linear", "quadratic"),
  conf_level = 0.95
)
```

Arguments

ratings	A units $\times$ raters matrix or data.frame. Rows = units; columns = raters. NA entries are allowed.
weights	One of "unweighted" (default; computes AC1) or "linear" / "quadratic" (compute AC2 with weight matrices used by weighted-κ conventions).
conf_level	Confidence level. Default 0.95.

Details

Coefficient.

$$\widehat{\mathrm{AC}} \;=\; \frac{p_a - p_e}{1 - p_e},$$

identical to Cohen's κ in structure but with a different chance-correction p_e:

$$p_e \;=\; \frac{T_w}{Q(Q-1)} \sum_{k=1}^{Q} \pi_k (1 - \pi_k),$$

where π_k is the mean within-unit proportion of category k, Q is the number of categories, and T_w is the sum of all entries of the weight matrix. For nominal data with unit weights (AC1), $T_w = Q$ and p_e reduces to $(1/(Q-1)) \sum_k \pi_k (1 - \pi_k)$.

Why AC over κ. κ can be near zero even when raters agree on almost every unit if the trait is rare or very common (the "kappa paradox"; Feinstein & Cicchetti, 1990). Gwet's AC keeps the same chance-correction logic but uses a less extreme reference distribution.

Variance. The SE is Gwet's (2008) linearization variance,

$$\mathrm{Var}(\widehat{\mathrm{AC}}) \;=\; \frac{1 - f}{n(n-1)} \sum_{i} (\widehat{\mathrm{AC}}_i^{*} - \widehat{\mathrm{AC}})^2,$$

where $\widehat{\mathrm{AC}}_i^{*}$ is the ith unit's influence value, combining its agreement and chance-term contributions, n is the number of units, and f is the sampling fraction (0 for an infinite target population). The interval is $\widehat{\mathrm{AC}} \pm t_{1-\alpha/2,\, n-1} SE$, with the upper limit truncated at 1. These quantities match Gwet's (2014) reference software.

Value

A `data.frame` with rows for the point estimate $\widehat{\mathrm{AC}}$, the standard error, the CI lower and upper limits, the percent agreement p_a, and the chance-agreement term p_e.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Feinstein, A. R., & Cicchetti, D. V. (1990). High agreement but low kappa: I. The problems of two paradoxes. *Journal of Clinical Epidemiology, 43*(6), 543–549. doi:10.1016/08954356(90)90158L

Gwet, K. L. (2008). Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology, 61*(1), 29–48. doi:10.1348/000711006X126600

Gwet, K. L. (2014). *Handbook of inter-rater reliability* (4th ed.). Advanced Analytics, LLC.

See Also

`cohen_kappa`, `fleiss_kappa`, `krippendorff_alpha`

Other agreement and measurement: `R2_mixed_effects()`, `content_validity_index()`, `icc_lmer()`, `krippendorff_alpha()`, `limits_of_agreement()`, `lin_ccc()`, `variance_components_mls()`

Examples

```
# 1. Unweighted AC1, two raters, nominal:
set.seed(113)
r1 <- sample(c("A", "B", "C"), 50, replace = TRUE)
r2 <- ifelse(runif(50) < 0.8, r1, sample(c("A", "B", "C"), 50, TRUE))
gwet_ac(cbind(r1, r2))

# 2. AC2 with linear weights, ordinal scale 1-5:
set.seed(113)
r1 <- sample(1:5, 60, replace = TRUE)
r2 <- pmin(5, pmax(1, r1 + sample(-1:1, 60, replace = TRUE)))
gwet_ac(cbind(r1, r2), weights = "linear")
```

---

helmert_coding *Helmert-Coding Contrast Matrix for a Factor*

Description

Builds the Helmert-coding contrast matrix for a factor with a levels. The k-th column contrasts the $(k+1)$-th level against the average of all preceding levels, giving a fully orthogonal set under equal sample sizes. The returned matrix has columns named after the contrasted level rather than the numeric column names produced by `stats::contr.helmert()`.

Usage

```
helmert_coding(levels)
```

Arguments

`levels`	Either an integer giving the number of levels or a character / factor vector giving the level labels. If integer, the labels default to `"L1"`, `"L2"`, ...

Details

Why Helmert. Helmert contrasts are the canonical "sequential" orthogonal contrast set: under equal-n, every column is orthogonal to every other column and to the intercept. They are useful when the factor has a natural ordering and the research questions are "does the k-th level differ from the average of the preceding levels?"

Equivalent to. `stats::contr.helmert()` but with interpretable column names.

Value

A numeric $a \times (a-1)$ matrix with row names = the factor levels and column names of the form `"L2_vs_prior"`, `"L3_vs_prior"`, ...

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (3rd ed.). Lawrence Erlbaum.

See Also

`contr.helmert`, `effects_coding`, `is_orthogonal_set`

Other design utilities: `design_consequences()`, `design_effect()`, `effects_coding()`, `is_orthogonal_set()`, `orthogonal_polynomial()`

Examples

```
# 1. Helmert coding for a 4-level factor:
helmert_coding(c("baseline", "week1", "week2", "week3"))

# 2. Confirm orthogonality:
M <- helmert_coding(4)
is_orthogonal_set(M)
```

---

holzinger_swineford *Holzinger and Swineford (1939) Factor Analysis Study*

Description

The complete data set from Holzinger and Swineford's (1939) *A study in factor analysis: The stability of a bi-factor solution*. Scores on 26 ability tests for 301 seventh and eighth grade pupils at two Chicago elementary schools, Pasteur (n = 156) and Grant-White (n = 145). The data have been used over the subsequent decades as one of the most-cited benchmarks in factor analysis, confirmatory factor analysis, structural equation modeling, and reliability research.

Usage

```
holzinger_swineford
```

Format

A data frame with 301 observations and 34 variables.

`id` Case identifier as in the original monograph. The numbering is not strictly consecutive; a small number of cases were dropped during data preparation and the original numbering was preserved (hence the visible skips).

`sex` Factor with levels `Female` and `Male`.

`grade` Grade in school, 7 or 8.

`age` Age in completed years, ignoring months past the most recent birthday.

`month_since_birthday` Completed months since the most recent birthday.

`age_months` Age in completed months, computed as `12 * age + month_since_birthday`.

`age_years` Age in fractional years, computed as `age + month_since_birthday / 12`.

`school` Factor with levels `Grant-White` and `Pasteur`, naming the two Chicago elementary schools from which pupils were drawn.

`t1_visual_perception` Visual perception test (spatial).

`t2_cubes` Cubes test (spatial).

`t3_paper_form_board` Paper form board test (spatial).

`t4_lozenges` Lozenges test (spatial).

`t5_general_information` General information test (verbal).

`t6_paragraph_comprehension` Paragraph comprehension test (verbal).

`t7_sentence` Sentence completion test (verbal).

`t8_word_classification` Word classification test (verbal).

`t9_word_meaning` Word meaning test (verbal).

`t10_addition` Addition test (mental speed).

`t11_code` Code test (mental speed).

`t12_counting_groups_of_dots` Counting groups of dots test (mental speed).

`t13_straight_and_curved_capitals` Straight and curved capitals test (mental speed).

`t14_word_recognition` Word recognition test (memory).

`t15_number_recognition` Number recognition test (memory).

`t16_figure_recognition` Figure recognition test (memory).

`t17_object_number` Object-number test (memory).

`t18_number_figure` Number-figure test (memory).

`t19_figure_word` Figure-word test (memory).

`t20_deduction` Deduction test (reasoning).

`t21_numerical_puzzles` Numerical puzzles test (reasoning).

`t22_problem_reasoning` Problem reasoning test (reasoning).

`t23_series_completion` Series completion test (reasoning).

`t24_woody_mccall` Woody-McCall mixed fundamentals, form I (arithmetic).

`t25_paper_form_board_r` Revised paper form board, administered only to the Grant-White pupils as an experimental substitute for `t3_paper_form_board`. `NA` for the 156 Pasteur pupils.

`t26_flags` Flags test, administered only to the Grant-White pupils as an experimental substitute for `t4_lozenges`. `NA` for the 156 Pasteur pupils.

Details

Karl John Holzinger (1893 to 1954) was a quantitative psychologist at the University of Chicago and one of the central figures in the first generation of factor analysis. He spent the 1922 to 1923 academic year working with Charles Spearman at University College London, absorbing Spearman's two-factor theory of intelligence, and later developed the *bi-factor model* as an extension of that theory. The bi-factor model posits a single general intelligence factor that runs through all tests,

plus several group factors that capture residual correlation among substantively related subgroups of tests. It is widely regarded as a precursor of modern hierarchical and orthogonal-bifactor models in psychometrics. Frances Swineford was Holzinger's research collaborator at the University of Chicago and a coauthor on much of his applied work.

The 1939 monograph reports a study of pupils in seventh and eighth grade classrooms at two Chicago elementary schools, Pasteur and Grant-White. Two schools were used deliberately, so that the stability of a bi-factor solution could be assessed by fitting the same model in each school and comparing the results. The 26 tests were designed to span five hypothesized ability domains:

- **Spatial**: tests 1 to 4 (visual perception, cubes, paper form board, lozenges), with tests 25 (a revised paper form board) and 26 (flags) administered only to the Grant-White sample as experimental substitutes for tests 3 and 4.
- **Verbal**: tests 5 to 9 (general information, paragraph comprehension, sentence completion, word classification, word meaning).
- **Mental speed**: tests 10 to 13 (addition, code, counting groups of dots, straight and curved capitals).
- **Memory**: tests 14 to 19 (word recognition, number recognition, figure recognition, object-number, number-figure, figure-word).
- **Reasoning and arithmetic**: tests 20 to 24 (deduction, numerical puzzles, problem reasoning, series completion, Woody-McCall mixed fundamentals).

Holzinger and Swineford concluded that the bi-factor solution was reasonably stable across the two schools, supporting the substantive interpretation of a general factor together with group factors.

The data have far outlived their original purpose. Jöreskog (1969) used a 9-test subset drawn from the Grant-White sample (n = 145) to introduce confirmatory maximum likelihood factor analysis; that 9-test subset is the version most modern confirmatory factor analysis tutorials use and is shipped in per-item-rescaled form as `HolzingerSwineford1939` in the **lavaan** package. The complete 26-test data shipped here support a wider range of analyses, including comparisons of the spatial, verbal, speed, memory, and reasoning ability blocks and multiple group analyses across the Pasteur and Grant-White schools.

The values in `holzinger_swineford` are the corrected version of the data, identical on all 26 test cells to `MBESS::HS` from MBESS version 4.9.3 onward and to `psychTools::holzinger.raw`. An older version of the data, with approximately 53 cell values that were later corrected, continues to circulate as `HS.data` in the **sem** package and as `HS.ability.data` in the **OpenMx** package; both are byte-identical snapshots taken from **MBESS** version 4.6.0 prior to the correction. The corrections are concentrated on the memory and reasoning tests, with the largest cluster on `t20_deduction` (15 cells, including a number of sign flips that reflect a corrected guessing-penalty adjustment).

Author(s)

Ken Kelley

Source

Holzinger, K. J., and Swineford, F. (1939). *A study in factor analysis: The stability of a bi-factor solution* (Supplementary Educational Monographs, No. 48). University of Chicago Press.

References

Holzinger, K. J., and Swineford, F. (1939). *A study in factor analysis: The stability of a bi-factor solution* (Supplementary Educational Monographs, No. 48). University of Chicago Press.

Jöreskog, K. G. (1969). A general approach to confirmatory maximum likelihood factor analysis. *Psychometrika, 34*, 183–202.

Holzinger, K. J. (1944). A simple method of factor analysis. *Psychometrika, 9*, 257–262.

Examples

```
data(holzinger_swineford)
str(holzinger_swineford)

# School and grade breakdown.
table(holzinger_swineford$school, holzinger_swineford$grade)

# Jöreskog (1969) drew nine tests from the Grant-White sample, and
# that subset became the modern confirmatory factor analysis
# benchmark.
joreskog_subset <- subset(
  holzinger_swineford,
  school == "Grant-White",
  select = c(t1_visual_perception, t2_cubes, t4_lozenges,
             t6_paragraph_comprehension, t7_sentence,
             t9_word_meaning, t10_addition,
             t12_counting_groups_of_dots,
             t13_straight_and_curved_capitals)
)
dim(joreskog_subset)
```

---

htmt *Heterotrait-Monotrait Ratio of Correlations (HTMT)*

Description

Computes the HTMT discriminant-validity index of Henseler, Ringle, and Sarstedt (2015) for every pair of constructs: the average correlation between items of *different* constructs, divided by the geometric mean of the average correlations among items *within* each construct. Two constructs whose HTMT approaches 1 are empirically indistinguishable however cleanly the model draws them; the customary red flags are 0.85 (strict) or 0.90 (liberal). An optional bootstrap gives a one-sided upper confidence bound, the quantity actually compared against the cutoff in the validity literature.

Usage

```
htmt(data, blocks, B = 0, conf_level = 0.95, seed = NULL)
```

Arguments

data	A data.frame of item responses.
blocks	Named list of character vectors: each element names a construct and gives its item columns (two or more per construct, two or more constructs).
B	Number of bootstrap resamples for the upper confidence bound; 0 (default) skips the bootstrap and reports the point estimates only.
conf_level	Confidence level for the one-sided upper bound. Defaults to 0.95.
seed	Optional integer seed for the bootstrap, used locally (the caller's random number generator state is restored on exit).

Details

All correlations are Pearson, computed on pairwise-complete observations. The statistic uses absolute average heterotrait correlations enter as absolute values, the convention of later implementations (the 2015 proposal used the plain correlations, which can cancel when signs mix; Roemer, Schuberth, & Henseler, 2021, recommend the absolute form for exactly that reason), and values near or above 1 indicating that the two item sets correlate across constructs about as strongly as within them. HTMT is a correlation-based screen, deliberately model-free; the confirmatory companion is the latent correlation between the two factors (see correction_for_attenuation and its factor-model discussion).

The bootstrap, when requested (B > 0), resamples the rows of data with replacement B times and recomputes every pairwise HTMT on each resample; the reported upper_limit is the conf_level empirical quantile of each pair's bootstrap distribution, a one-sided upper percentile bound (Efron & Tibshirani, 1993). That bound is the only interval offered, matching how the validity literature uses HTMT (the question is whether the ratio credibly exceeds the cutoff); no two-sided or bias-corrected and accelerated (BCa) variant is provided. A resample in which some block's average within-construct correlation is not positive leaves HTMT undefined there; such resamples are dropped, a single warning reports how many, and the bound is computed from the resamples that remained (the call stops only when fewer than 100 remain). Bootstrap results vary from run to run; supply seed for reproducibility.

Value

A data.frame (class dmar_tbl) with one row per construct pair: construct_1, construct_2, htmt, and, when B > 0, upper_limit (the one-sided conf_level bootstrap percentile bound).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, NY: Chapman & Hall/CRC.

Henseler, J., Ringle, C. M., & Sarstedt, M. (2015). A new criterion for assessing discriminant validity in variance-based structural equation modeling. *Journal of the Academy of Marketing Science, 43*(1), 115–135. doi:10.1007/s1174701404038

See Also

average_variance_extracted for the convergent side of the validity ledger; reliability_omega for the composite reliability of each block; correction_for_attenuation for the latent correlation route.

Other multivariate and latent variable methods: average_variance_extracted(), bifactor_indices(), cfa_1(), cfa_2(), cfa_k(), ci_eigenvalue(), common_method_marker(), common_method_single_factor(), dmacs(), ecvi(), irt_grm(), irt_information(), measurement_alignment(), measurement_invariance(), procrustes_phi(), simple_structure()

Examples

```
# Two clean constructs and their six items.
set.seed(113)
n <- 300
f1 <- rnorm(n); f2 <- 0.3 * f1 + sqrt(1 - 0.09) * rnorm(n)
d <- data.frame(
  a1 = .8 * f1 + rnorm(n, 0, .6), a2 = .7 * f1 + rnorm(n, 0, .7),
  a3 = .6 * f1 + rnorm(n, 0, .8),
  b1 = .8 * f2 + rnorm(n, 0, .6), b2 = .7 * f2 + rnorm(n, 0, .7),
  b3 = .6 * f2 + rnorm(n, 0, .8))
h <- htmt(d, blocks = list(A = c("a1", "a2", "a3"),
                           B = c("b1", "b2", "b3")))
h

# The broom verbs: one row per construct pair.
generics::tidy(h)
generics::glance(h)

# The upper confidence bound, which is the quantity the validity
# literature compares against 0.85 or 0.90, comes from a bootstrap
# that recomputes every pairwise ratio on each of B resamples; the
# table gains an upper_limit column. B = 1000 keeps the example
# quick; a claim about discriminant validity deserves the bound at
# B = 10000 rather than the point estimate alone.
htmt(d, blocks = list(A = c("a1", "a2", "a3"),
                      B = c("b1", "b2", "b3")),
     B = 1000, seed = 113)
```

---

icc *Intraclass Correlation Coefficients With Confidence Intervals*

Description

Computes one or more of the six standard intraclass correlation coefficients (ICC) of Shrout and Fleiss (1979) for an $n \times k$ matrix of ratings (rows = subjects, columns = raters/measurements), along with the F-distribution-based confidence interval for each.

Usage

```
icc(x, type = "ICC(2,1)", conf_level = 0.95)
```

Arguments

`x`	An $n \times k$ numeric matrix or `data.frame`: each row is a subject (target) and each column is a rater (or repeated measurement). Missing values are not supported; remove or impute first.
`type`	Which ICC variant(s) to return. Aliases include `"1"`, `"2"`, `"3"` (single rater versions of types 1, 2, 3) and `"1k"`, `"2k"`, `"3k"` (average-of-k-raters versions). The full names `"ICC(1,1)"`, `"ICC(2,1)"`, `"ICC(3,1)"`, `"ICC(1,k)"`, `"ICC(2,k)"`, `"ICC(3,k)"` are also accepted, as is `"all"`. Vectors are accepted; a row is returned for each type.
`conf_level`	Confidence level for the interval (default `0.95`).

Details

The six variants follow Shrout and Fleiss's (1979) classification:

- `ICC(1,1)`: one-way random model, single rater. Each subject is rated by a different (random) rater; appropriate when raters are not crossed with subjects.
- `ICC(1,k)`: one-way random model, average of k raters.
- `ICC(2,1)`: two-way random model, single rater, absolute agreement. Subjects $\times$ raters fully crossed; both effects random.
- `ICC(2,k)`: two-way random model, average of k raters, absolute agreement.
- `ICC(3,1)`: two-way mixed model, single rater, consistency. Raters are fixed (the only ones of interest); rates differences in mean across raters are not penalized.
- `ICC(3,k)`: two-way mixed model, average of k raters, consistency.

Confidence intervals follow the F-distribution-based formulas of Shrout and Fleiss (1979, pp.\ 425–426); the `ICC(2,1)` interval uses their asymmetric approximate-*df* formulation.

Value

A `data.frame` (class `dmar_tbl`) with one row per requested ICC type and columns `type`, `value` (point estimate), `lower_limit`, `upper_limit`, `F_value`, `df_1`, `df_2`, and `p_value` (for the implicit null $H_0 : \mathrm{ICC} = 0$).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Shrout, P. E., & Fleiss, J. L. (1979). Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin, 86*(2), 420–428.

McGraw, K. O., & Wong, S. P. (1996). Forming inferences about some intraclass correlation coefficients. *Psychological Methods, 1*(1), 30–46. doi:10.1037/1082989X.1.1.30

See Also

ci_r, descriptives

Other reliability: cohen_kappa(), diagnosis_agreement, fleiss_kappa(), reliability(), reliability_H(), reliability_alpha(), reliability_kr20(), reliability_omega(), reliability_omega_categor

Examples

```
# Shrout & Fleiss (1979), Table 1: 4 raters, 6 targets.
shrout_fleiss <- matrix(
  c(9, 2, 5, 8,
    6, 1, 3, 2,
    8, 4, 6, 8,
    7, 1, 2, 6,
    10, 5, 6, 9,
    6, 2, 4, 7),
  nrow = 6, byrow = TRUE,
  dimnames = list(paste0("Target_", 1:6),
                  paste0("Judge_",  1:4))
)

icc(shrout_fleiss, type = "all")

# Just the two-way mixed-model single-rater consistency ICC:
icc(shrout_fleiss, type = "ICC(3,1)")
```

---

icc_lmer *Intraclass Correlation From a Fitted* lme4 *Mixed-Effects Model*

Description

Reads the variance-component decomposition off a fitted lmer model and returns the implied intraclass correlation (ICC) for the specified grouping factor along with a Bonett (2002) Fisher-L-transform confidence interval, in tidy long form. The bridge function between DMAR's classical-ANOVA path (variance_components_mls, var_icc) and the modern mixed-effects path.

Usage

```
icc_lmer(fit, group = NULL, conf_level = 0.95)
```

Arguments

fit	A fitted lmer model (class lmerMod) with at least one random-effects grouping factor.
group	Character name of the grouping factor whose ICC is wanted. If NULL (default), the first random-effects grouping factor is used. For three-level models, supply the target level explicitly.
conf_level	Confidence level. Default 0.95.

Details

Definition. For a two-level model with random intercept by group j and within-group residual,

$$\rho \;=\; \frac{\sigma_b^2}{\sigma_b^2 + \sigma_w^2},$$

read directly from `VarCorr(fit)`. For three-level models, the user specifies which level (`group`) provides the variance contribution; the denominator is the total variance summed across all variance components.

Bonett (2002) CI. The CI is built on the Fisher-style L-transformation

$$L \;=\; \tfrac{1}{2}\log\left(\frac{1+(k-1)\rho}{1-\rho}\right),$$

with variance $k/(2(k-1)(n-2))$, where k is the average cluster size and n is the number of clusters. Back- transformation keeps the bounds in $[0, 1]$.

Limitations. The Bonett CI is built on a balanced / approximately balanced approximation; for severely unbalanced designs the profile likelihood CI from `confint(fit, ...)` is preferable.

Value

A `data.frame` with rows for the point estimate of the ICC, the variance components (between-group and residual), the implied total variance, the Bonett (2002) CI lower/upper limits, and the cluster-level effective sample size used in the CI.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bonett, D. G. (2002). Sample size requirements for estimating intraclass correlations with desired precision. *Statistics in Medicine, 21*(9), 1331–1335. doi:10.1002/sim.1108

Donner, A. (1986). A review of inference procedures for the intraclass correlation coefficient in the one-way random effects model. *International Statistical Review, 54*(1), 67–82.

Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel analysis: An introduction to basic and advanced multilevel modeling* (2nd ed.). Sage.

See Also

`icc`, `var_icc`, `variance_components_mls`, `ss_aipe_icc`, `lmer`

Other agreement and measurement: `R2_mixed_effects()`, `content_validity_index()`, `gwet_ac()`, `krippendorff_alpha()`, `limits_of_agreement()`, `lin_ccc()`, `variance_components_mls()`

Other mixed models: `R2_mixed_effects()`, `R2_mixed_effects_decomposition()`, `manova_split_plot()`, `mixed_anova()`, `ss_aipe_mixed_effects()`, `ss_aipe_mixed_effects_sensitivity()`, `ss_power_mixed_effects()`, `ss_power_split_plot_anova()`

Examples

```
# Twenty groups of six, with a group-level standard deviation of 0.7 on
# top of within-group noise with a standard deviation of 1, so the data
# generating ICC is 0.7^2 / (0.7^2 + 1) = 0.329. The estimate below comes
# with a wide interval: 20 clusters is not many, and the interval is what
# keeps that fact visible.
set.seed(113)
n_grp <- 20; n_per <- 6
grp <- factor(rep(1:n_grp, each = n_per))
y   <- rnorm(n_grp * n_per, 0, 1) + rep(rnorm(n_grp, 0, 0.7), each = n_per)
d <- data.frame(y, grp)
fit <- lme4::lmer(y ~ 1 + (1 | grp), data = d)
icc_lmer(fit)
```

---

irt_grm *Graded Response Model for Ordered Categorical Items*

Description

Estimates Samejima's (1969) graded response model for a set of ordered categorical items (Likert items, symptom severity ratings, rubric scored performance items) and reports each item's discrimination and its category boundary locations. The model is fitted as the single-factor categorical factor analysis model it provably is (Takane and de Leeuw, 1987), using **lavaan**'s categorical estimator on the polychoric correlations, and the solution is then converted to the normal ogive (or logistic) item response theory parameterization. A researcher who already fits confirmatory factor analysis models therefore gets item response theory item parameters without adopting a second estimation engine, and the two analyses of the same items stay in one modeling tradition.

Usage

```
irt_grm(
  data,
  items = NULL,
  estimator = "WLSMV",
  metric = c("normal_ogive", "logistic")
)
```

Arguments

data	A `data.frame` or matrix of ordered item responses, one column per item, coded with integer category values (for example 1, 2, 3, 4, 5). Rows with any missing value on the analyzed items are listwise-deleted. Each item must have at least 2 and at most 20 distinct observed categories; a column with more than 20 distinct values, or with non-integer values, is treated as continuous and rejected.
items	Optional character vector naming the columns of `data` to analyze. Defaults to `NULL`, which uses every column. At least 3 items are required.

`estimator` Character; the **lavaan** estimator for categorical data. The choices differ in the weight matrix applied to the polychoric correlations and in whether the test statistic is corrected. `"WLSMV"` (the default) is diagonally weighted least squares with a mean- and variance-adjusted test statistic (Muthén, 1984; Muthén, du Toit, & Spisic, 1997), the standard estimator for ordered categorical items; `"WLSM"` applies the mean adjustment only. `"DWLS"` is the same diagonal-weight estimator with no correction. `"WLS"` uses the full weight matrix (the asymptotic distribution free approach; Browne, 1984), which is unstable unless the sample is large relative to the number of thresholds. `"ULS"`, unweighted least squares, uses an identity weight matrix, an option worth considering in small samples where even the diagonal weights are noisy; `"ULSMV"` and `"ULSM"` add the corrected test statistics. When in doubt keep the default.

`metric` Which discrimination metric to report in the `a` column: `"normal_ogive"` (default) or `"logistic"`. Both are always computed and the one not reported in `a` is attached as an attribute, so the returned table has the same columns and the same number of rows either way. The boundary locations `b` are identical under the two metrics.

Details

One model, two parameterizations. Samejima's (1969) graded response model and the single-factor categorical factor analysis model of Muthén (1984) are the same model written in different parameterizations; Takane and de Leeuw (1987) proved the equivalence, and Kamata and Bauer (2008) give the algebra item by item. Each observed response X_i is a categorization of a latent continuous response variate X_i^* at thresholds τ_{ik}, and $X_i^* = \lambda_i\theta + \varepsilon_i$ with θ standard normal and X_i^* standardized. Fitting that model on the polychoric correlations and converting the solution gives the normal ogive graded response model directly. For item i with standardized loading λ_i and standardized thresholds τ_{ik},

$$a_i = \frac{\lambda_i}{\sqrt{1-\lambda_i^2}}, \qquad b_{ik} = \frac{\tau_{ik}}{\lambda_i}.$$

The boundary response function is

$$P_{ik}^*(\theta) = \Phi[a_i(\theta - b_{ik})]\,,$$

the probability of responding above boundary k, with $P_{i0}^*(\theta) \equiv 1$ and $P_{iK}^*(\theta) \equiv 0$; the probability of the individual category is the difference of adjacent boundary functions, $P_{ik}(\theta) = P_{i,k-1}^*(\theta) - P_{ik}^*(\theta)$. When $a_i > 0$, that is, for an item keyed in the same direction as the rest of the scale, P_{ik}^* is monotone increasing in θ, the boundary locations of the item are ordered, $b_{i1} < b_{i2} < \cdots$, and b_{ik} is the value of θ at which the probability of responding above boundary k reaches 0.50. An item keyed in the opposite direction has $\lambda_i < 0$, hence $a_i < 0$ and boundary locations that run from high to low; see the two paragraphs on direction below.

The direction of the latent variable. A single-factor model fixes θ only up to its direction. Relabeling θ as $-\theta$ changes the sign of every loading and leaves the fitted model, the thresholds, and every fit measure exactly as they were, so it is a renaming of the latent direction rather than a different model. The thresholds are untouched because τ_{ik} cuts the item's own latent response variate X_i^*, which the relabeling does not move; the sign change therefore passes straight through to $a_i = \lambda_i/\sqrt{1-\lambda_i^2}$ and to $b_{ik} = \tau_{ik}/\lambda_i$, both of which change sign. **lavaan** returns whichever direction its starting values point toward, and for a scale that contains a reverse-keyed item that

direction can turn on something as incidental as the order of the columns. The solution is therefore put in a fixed direction before it is converted: if the standardized loadings sum to a negative number the whole factor is flipped, so that θ runs in the direction the scale as a whole measures. The result is the same table no matter how the columns are ordered. Whether the flip was applied is recorded on the `"factor_sign_flipped"` attribute. The **lavaan** object on the `"fit"` attribute is the fit as **lavaan** produced it, so when a flip was applied its loadings carry the opposite sign to the `lambda` column.

Reverse-keyed items. An item whose loading is still negative after the direction is fixed is keyed opposite to the rest of the scale, which is a property of the item rather than an artifact of the sign indeterminacy. Its discrimination is negative and its boundary locations run from high to low, so it does not satisfy the graded response model as written above and its parameters do not belong on the same scale as the others. Such items are named in a warning. Reverse score them (for example `x <- (min(x) + max(x)) - x`) and refit; that puts the item in the direction the rest of the scale measures and restores $a_i > 0$ and the ordering $b_{i1} < b_{i2} < \cdots$.

The two discrimination metrics and the constant 1.702. The conversion above puts a_i in the normal ogive metric, where the boundary function is a normal cumulative distribution function. The item response theory literature more often writes the graded response model with a logistic boundary function, and the two agree closely once the logistic argument is stretched by a scaling constant: $|\Phi(x) - \Psi(1.702x)| < 0.01$ for every x, where Ψ is the standard logistic cumulative distribution function. The value 1.702 is the constant that minimizes that maximum discrepancy (Haley, 1952; see Camilli, 1994, for the history), so $a_i(\text{logistic}) = 1.702\, a_i(\text{normal ogive})$ and software that reports logistic slopes (for example **mirt** and the classical two parameter logistic tradition) gives values about 1.7 times larger for the same items. The scaling multiplies the slope and leaves the location alone, so b_{ik} does not depend on the metric.

Estimation and what to expect. lavaan estimates the thresholds and the polychoric correlations, then fits the single-factor model to those correlations by (diagonally) weighted least squares. This is limited information estimation: it uses the univariate and bivariate margins of the response table, whereas marginal maximum likelihood (the usual item response theory approach, as in **mirt**) uses the full response pattern likelihood. The two are consistent for the same population parameters and agree closely in practice, but they are different estimators and will not return identical numbers on a finite sample. Limited information estimation scales well to many items and brings the whole apparatus of factor analysis fit assessment (CFI, TLI, RMSEA) along with it; the fit measures are returned on the `"fit_measures"` attribute and the **lavaan** object itself on `"fit"`, so any **lavaan** accessor can be applied to the result.

The model is unidimensional by construction. A standardized loading at or beyond one is an improper (Heywood) solution: the implied discrimination is infinite and the conversion is not interpretable. That case is flagged with a warning rather than silently returned as a number.

This function requires **lavaan** to be installed.

Value

A `data.frame` (class `dmar_tbl`) with one row per item and category boundary and the columns

`item` Item name, taken from the column name.

`factor` Name of the latent variable, the same for every row in this unidimensional model.

`category` Boundary index k, running from 1 to one fewer than the item's number of categories.

`lambda` Standardized factor loading λ_i of the item's latent response variate on the factor, repeated across the item's boundaries.

`tau` Standardized threshold τ_{ik}.

`a` Discrimination in the metric named by `metric`, repeated across the item's boundaries.

`b` Boundary location b_{ik} on the θ scale.

The attributes are `"fit"` (the fitted **lavaan** object), `"fit_measures"` (the full named numeric vector from `lavaan::fitMeasures`, unrounded), `"metric"` (the reported discrimination metric), `"estimator"`, `"n_categories"` (named integer vector of the number of observed categories per item), `"N"` (the analyzed sample size), `"factor_sign_flipped"` (a single logical recording whether the direction of the latent variable was reversed to satisfy the sign convention described in Details), and whichever of `"a_logistic"` or `"a_normal_ogive"` was not reported in the `a` column (a named numeric vector, one element per item).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Camilli, G. (1994). Teacher's corner: Origin of the scaling constant $d = 1.7$ in item response theory. *Journal of Educational and Behavioral Statistics, 19*(3), 293–295. doi:10.3102/10769986019003293

Haley, D. C. (1952). *Estimation of the dosage mortality relationship when the dose is subject to error* (Technical Report No. 15). Applied Mathematics and Statistics Laboratory, Stanford University.

Kamata, A., & Bauer, D. J. (2008). A note on the relation between factor analytic and item response theory models. *Structural Equation Modeling, 15*(1), 136–153. doi:10.1080/10705510701758406

Muthén, B. (1984). A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators. *Psychometrika, 49*(1), 115–132.

Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph Supplement, 34*(4, Pt. 2), 1–97.

Takane, Y., & de Leeuw, J. (1987). On the relationship between item response theory and factor analysis of discretized variables. *Psychometrika, 52*(3), 393–408.

Wirth, R. J., & Edwards, M. C. (2007). Item factor analysis: Current approaches and future directions. *Psychological Methods, 12*(1), 58–79. doi:10.1037/1082989X.12.1.58

See Also

`cfa_1` (the same single-factor model reported in the factor analysis parameterization), `reliability_omega_categorical` (reliability for the same class of items), `cfa`.

Other multivariate and latent variable methods: `average_variance_extracted()`, `bifactor_indices()`, `cfa_1()`, `cfa_2()`, `cfa_k()`, `ci_eigenvalue()`, `common_method_marker()`, `common_method_single_factor()`, `dmacs()`, `ecvi()`, `htmt()`, `irt_information()`, `measurement_alignment()`, `measurement_invariance()`, `procrustes_phi()`, `simple_structure()`

Examples

```
# Six five-category items generated from a known graded response model.
set.seed(113)
n <- 800
a_pop <- c(1.2, 0.9, 1.5, 1.0, 1.3, 1.1)
b_pop <- rbind(c(-1.6, -0.6, 0.3, 1.2), c(-1.4, -0.4, 0.5, 1.5),
               c(-1.8, -0.7, 0.2, 1.1), c(-1.2, -0.2, 0.7, 1.6),
               c(-1.5, -0.5, 0.4, 1.3), c(-1.3, -0.3, 0.6, 1.4))
theta <- rnorm(n)
responses <- vapply(seq_along(a_pop), function(i) {
  p_star <- outer(theta, b_pop[i, ], function(z, b) pnorm(a_pop[i] * (z - b)))
  as.integer(1 + rowSums(runif(n) < p_star))
}, integer(n))
colnames(responses) <- paste0("item", seq_along(a_pop))
responses <- as.data.frame(responses)

# Item parameters in the normal ogive metric.
grm <- irt_grm(responses)
grm

# The generating discriminations, for comparison.
a_pop

# Model fit travels with the item parameters.
attr(grm, "fit_measures")[c("cfi", "tli", "rmsea", "srmr")]

# The same fit reported with logistic slopes, about 1.702 times the
# normal ogive slopes above; the boundary locations do not change.
irt_grm(responses, metric = "logistic")

# A subset of the items, selected by name, is fit on its own.
irt_grm(responses, items = c("item1", "item3", "item5"))

# The boundary response function of the first item at theta = 0.
first <- grm[grm$item == "item1", ]
pnorm(first$a * (0 - first$b))
```

`irt_information` *Item and Test Information for the Graded Response Model*

Description

Evaluates the item information functions and the test information function of a graded response model on a grid of latent trait values, together with the standard error of the latent trait estimate, $SE(\theta) = 1/\sqrt{I(\theta)}$. Reliability is a single number that describes a scale at one place on the latent continuum; the information function is the same idea expressed as a function of where the respondent sits, so an item pool can be judged on where it measures precisely rather than on one global summary. Because information is additive across items, the curve also shows which items

carry the precision, and over what range, which is what makes it useful for building and trimming a scale.

Usage

```
irt_information(
  a,
  b = NULL,
  item = NULL,
  theta = seq(-4, 4, length.out = 81),
  grm = NULL
)
```

Arguments

a	Discriminations, a numeric vector of positive values. Supply either one value per item (named with the item names, or in the order the items first appear in item) or one value per boundary row (constant within an item). Not used when grm is supplied.
b	Boundary locations (category thresholds), a numeric vector with one element per category boundary. An item with m categories has $m - 1$ boundaries, which the model requires to be in ascending order within the item. Not used when grm is supplied.
item	Item labels, a character or factor vector the same length as b naming the item each boundary belongs to. When NULL (default), each element of b is treated as its own dichotomous item, named item_1, item_2, and so on, and a must then have the same length as b. Not used when grm is supplied.
theta	Latent trait values at which to evaluate the information functions. Any finite numeric vector; the default, seq(-4, 4, length.out = 81), covers the range in which almost all of a standard normal trait distribution falls, in steps of 0.1.
grm	Optionally, the result of irt_grm(): a data.frame with one row per item and category boundary and columns item, a, and b (a category column, when present, orders the boundaries within an item). Supply this or the parameters, not both.

Details

For item *i* with discrimination a_i and ordered boundary locations $b_{i1} < b_{i2} < \cdots < b_{i,m-1}$ for *m* categories, the normal ogive graded response model of Samejima (1969) defines the boundary response function

$$P^*_{ik}(\theta) = \Phi[a_i(\theta - b_{ik})],$$

the probability of responding *above* boundary *k*, that is, in any category higher than the *k*th, with the conventions $P^*_{i0} = 1$ and $P^*_{im} = 0$. The category response function is the difference of adjacent boundary functions,

$$P_{ik}(\theta) = P^*_{i,k-1}(\theta) - P^*_{ik}(\theta),$$

and differentiating with respect to θ gives

$$P'_{ik}(\theta) = a_i\{\phi[a_i(\theta - b_{i,k-1})] - \phi[a_i(\theta - b_{ik})]\},$$

where ϕ is the standard normal density and the density terms vanish at the two extreme categories (there is no b_{i0} and no b_{im}). Item information is

$$I_i(\theta) = \sum_{k=1}^{m} \frac{[P'_{ik}(\theta)]^2}{P_{ik}(\theta)},$$

test information is $I(\theta) = \sum_i I_i(\theta)$, and the standard error of the maximum likelihood estimate of θ is $SE(\theta) = 1/\sqrt{I(\theta)}$.

Two properties make the curve worth reading. Information is additive across items, so an item's contribution can be read off directly and a pool can be assembled to cover a targeted range. And the reciprocal relation to the squared standard error means the peak of the curve locates where the scale estimates the trait most precisely, reported here as the `"theta_max_information"` attribute.

For a dichotomous item the model reduces to the two parameter normal ogive, whose information has the closed form

$$I_i(\theta) = \frac{a_i^2 \phi[a_i(\theta - b_i)]^2}{\Phi[a_i(\theta - b_i)]\{1 - \Phi[a_i(\theta - b_i)]\}},$$

which the general expression above reproduces; that identity is one of the tests of this function.

The category probabilities underflow to zero for θ far from every boundary, where the ratio $(P')^2/P$ would be $0/0$. A category whose probability is not strictly positive contributes zero to the sum, which is the limit the ratio approaches, so the returned information is finite and nonnegative on any grid, however extreme, and is never `NaN`. In the regime where $(P')^2$ underflows but P does not, the ratio is formed as $\exp[2 \log |P'| - \log P]$ so the contribution is kept rather than flushed to zero. Where two boundaries of an item coincide, the category between them has probability zero everywhere and, by the same guard, contributes nothing.

The parameters are in the normal ogive metric, which is what `irt_grm()` returns by default. The logistic metric used by much of the item response theory software scales the discrimination by approximately 1.702 (Camilli, 1994); a logistic a is put on the normal ogive scale by dividing by that constant. The two metrics give information functions that are proportional in shape but not equal in value, so a cross-software comparison is a comparison of curves, not of numbers.

Value

A `data.frame` (class `dmar_tbl`) with one row per value of `theta` and columns:

`theta` The latent trait value, as supplied.

`test_information` Test information at that value, the sum of the item information functions.

`se` The standard error of the latent trait estimate, $1/\sqrt{I(\theta)}$. It is `Inf` where test information is zero, which is the correct statement that the items carry no information there.

The result carries these attributes:

`"item_information"` A numeric matrix of item information with `theta` in the rows (row names are the `theta` values) and items in the columns (column names are the item names). Its row sums are `test_information`.

`"item"` The item names, in the order they appear in the columns of `"item_information"`.

`"a"` The discrimination used for each item, a numeric vector named by item.

`"b"` The boundary locations used, a numeric vector in item order and, within an item, in ascending order, named by the item each boundary belongs to.

`"theta_max_information"` The value of `theta` at which test information peaks on the supplied grid (the first such value if there are ties). It is a grid value, not the result of an optimization, so a finer `theta` locates the peak more sharply.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Baker, F. B., & Kim, S.-H. (2004). *Item response theory: Parameter estimation techniques* (2nd ed.). Marcel Dekker.

Camilli, G. (1994). Teacher's corner: Origin of the scaling constant $d = 1.7$ in item response theory. *Journal of Educational and Behavioral Statistics, 19*(3), 293–295. doi:10.3102/10769986019003293

Embretson, S. E., & Reise, S. P. (2000). *Item response theory for psychologists*. Lawrence Erlbaum.

Lord, F. M. (1980). *Applications of item response theory to practical testing problems*. Lawrence Erlbaum.

Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph Supplement, 34*(4, Pt. 2), 1–97.

See Also

`plot_irt_information` for the curve, `reliability_omega` for the single-number companion.

Other multivariate and latent variable methods: `average_variance_extracted()`, `bifactor_indices()`, `cfa_1()`, `cfa_2()`, `cfa_k()`, `ci_eigenvalue()`, `common_method_marker()`, `common_method_single_factor()`, `dmacs()`, `ecvi()`, `htmt()`, `irt_grm()`, `measurement_alignment()`, `measurement_invariance()`, `procrustes_phi()`, `simple_structure()`

Examples

```
# Three items: a five-category rating item and two dichotomous items.
# The discriminations are named, so they are matched to the item labels.
info <- irt_information(
  a = c(mood_1 = 1.4, mood_2 = 0.9, mood_3 = 1.1),
  b = c(-1.5, -0.5, 0.5, 1.5, 0.0, 0.8),
  item = c(rep("mood_1", 4), "mood_2", "mood_3")
)
head(info)

# Where does this three-item set measure most precisely?
attr(info, "theta_max_information")

# Each item's contribution; the rows sum to the test information.
head(attr(info, "item_information"))

# A dichotomous item matches the two parameter normal ogive closed form.
one <- irt_information(a = 1.5, b = 0.25, theta = c(-1, 0, 1))
```

```
z <- 1.5 * (c(-1, 0, 1) - 0.25)
1.5^2 * dnorm(z)^2 / (pnorm(z) * (1 - pnorm(z)))
one$test_information
```

`is_orthogonal_set` *Check Whether a Set of Contrasts Is Mutually Orthogonal*

Description

Tests whether every pair of columns in a contrast-coefficient matrix is orthogonal under either the equal-n convention $\sum_i c_{ik}c_{ij} = 0$ or the unequal-n convention $\sum_i c_{ik}c_{ij}/n_i = 0$ (Maxwell, Delaney, & Kelley, 2027, Sec. 4.10; Kirk, 2013). Also checks that each column sums to zero (the contrast property).

Usage

```
is_orthogonal_set(contrasts, n = NULL, tol = 1e-08)
```

Arguments

`contrasts`	A numeric $a \times m$ matrix or `data.frame`, where a is the number of groups and m is the number of contrasts.
`n`	Optional integer vector of length a giving the per- group sample sizes. If supplied, the unequal-n convention is used; otherwise the equal-n convention is assumed.
`tol`	Numerical tolerance for declaring orthogonality. Default `1e-8`.

Details

Equal-n. Two contrasts $\mathbf{c}, \mathbf{d}$ on a groups of equal size are orthogonal iff $\sum_{i=1}^{a} c_i d_i = 0$.

Unequal-n. With sample sizes $n_1, \ldots, n_a$, the orthogonality condition that yields uncorrelated sample contrasts is $\sum_{i=1}^{a} c_i d_i / n_i = 0$.

Useful for design checks. Before performing planned comparisons or partitioning the omnibus sums of squares, the user typically wants confirmation that the chosen contrast set is orthogonal so that its component SS sum to the omnibus SS.

Value

A `data.frame` with rows for the overall orthogonality flag (`1` = all pairs orthogonal, `0` = not), the contrast-sum-to-zero flag, the number of contrasts tested, and one row per pairwise dot-product, named by contrast pair.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kirk, R. E. (2013). *Experimental design: Procedures for the behavioral sciences* (4th ed.). Sage.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Sec. 4.10.)

See Also

effects_coding, helmert_coding, ci_scheffe

Other design utilities: design_consequences(), design_effect(), effects_coding(), helmert_coding(), orthogonal_polynomial()

Examples

```
# 1. Two orthogonal contrasts on a 4-group design (equal n):
cmat <- cbind(
  c_linear  = c(-3, -1,  1,  3),
  c_quad    = c( 1, -1, -1,  1)
)
is_orthogonal_set(cmat)

# 2. Same contrasts under unequal sample sizes:
is_orthogonal_set(cmat, n = c(10, 8, 12, 9))

# 3. Non-orthogonal pair:
cmat_bad <- cbind(
  c_diff_1  = c( 1, -1,  0,  0),
  c_diff_2  = c( 1,  0, -1,  0)
)
is_orthogonal_set(cmat_bad)
```

---

krippendorff_alpha *Krippendorff's α Inter-Rater Agreement*

Description

Computes Krippendorff's (1980, 2004, 2011) α, the most general chance-corrected inter-rater agreement coefficient. Unlike cohen_kappa (two raters, nominal data) or fleiss_kappa (multiple raters, nominal data), Krippendorff's α supports any number of raters, missing values, and any of four levels of measurement (nominal, ordinal, interval, ratio) via a user-specified distance metric.

Usage

```
krippendorff_alpha(
  ratings,
  level = c("nominal", "ordinal", "interval", "ratio"),
  conf_level = 0.95,
  boot = FALSE,
```

```
  B = 1000L,
  seed = NULL
)
```

Arguments

ratings	A units × raters matrix (or data.frame). Rows = units of analysis; columns = raters. NA entries are allowed.
level	One of "nominal" (default), "ordinal", "interval", or "ratio"; controls the distance metric used to compute disagreement.
conf_level	Confidence level for the bootstrap CI. Default 0.95.
boot	Logical. If TRUE, returns a bootstrap percentile CI. Set to FALSE to return only the point estimate (much faster).
B	Number of bootstrap resamples when boot = TRUE. Default 1000L.
seed	Optional integer seed for reproducibility of the bootstrap. Default NULL, which leaves the user's current RNG state intact; supply an integer for reproducibility.

Details

Coefficient. Krippendorff's α is

$$\alpha \;=\; 1 - \frac{D_o}{D_e},$$

where D_o is the observed disagreement (average squared distance over all within-unit pairs of ratings, scaled by the number of pairable values), and D_e is the expected disagreement (average squared distance over all between-unit pairs). The metric used in the squared distance depends on level:

- *nominal*: $d(a, b) = \mathrm{I}(a \neq b)$
- *ordinal*: distance based on cumulative rank counts
- *interval*: $d(a, b) = (a - b)^2$
- *ratio*: $d(a, b) = ((a - b)/(a + b))^2$

CI. The CI is by case-resampling bootstrap over units (rows): the rows of ratings are resampled with replacement B times and α is recomputed on each resample, so units are the sampling unit and the rater panel is treated as fixed. Only the percentile interval is offered: the limits are the empirical quantiles of the bootstrap estimates (Efron & Tibshirani, 1993); there is no bias-corrected and accelerated (BCa) variant. Resamples on which the coefficient cannot be computed (for example, a resample without enough pairable values) are dropped; the interval is computed from the ones that return a finite value, and how many did is reported as the B_used row of the result. No closed-form sampling variance is in general use for Krippendorff's alpha across its measurement levels and missing data patterns, so the bootstrap is the interval Krippendorff recommends (Krippendorff, 2011; Hayes & Krippendorff, 2007). B = 1000L typically gives a stable CI to two decimal places. The bootstrap is opt-in (boot = FALSE by default, which returns the point estimate alone and is much faster); ask for it whenever the coefficient is being reported rather than explored, since a point estimate on its own says nothing about how precisely α is determined. Bootstrap results vary from run to run; supply seed for reproducibility.

Interpretation. α ranges from $-D_e/D_o$ (perfect disagreement) through 0 (chance level) to 1 (perfect agreement). Report the coefficient with its confidence interval and judge it against the reliability the application requires; Krippendorff (2004) discusses how that judgment depends on the cost of acting on unreliable data.

Value

A `data.frame` with rows for the point estimate $\hat{\alpha}$, the observed disagreement D_o, the expected disagreement D_e, the number of pairable values, and, when a bootstrap was run, the lower and upper bootstrap CI limits and `B_used`, the number of resamples that returned a finite value and so entered the interval.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, NY: Chapman & Hall/CRC.

Hayes, A. F., & Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures, 1*(1), 77–89. doi:10.1080/19312450709336664

Krippendorff, K. (1980). *Content analysis: An introduction to its methodology*. Sage.

Krippendorff, K. (2004). *Content analysis: An introduction to its methodology* (2nd ed.). Sage.

Krippendorff, K. (2011). Computing Krippendorff's alpha-reliability. *Departmental Papers (ASC)*, Annenberg School for Communication, University of Pennsylvania.

See Also

`cohen_kappa`, `fleiss_kappa`, `icc`

Other agreement and measurement: `R2_mixed_effects()`, `content_validity_index()`, `gwet_ac()`, `icc_lmer()`, `limits_of_agreement()`, `lin_ccc()`, `variance_components_mls()`

Examples

```
# 1. Nominal ratings, 4 raters, 12 units (from Krippendorff 2011 Tab. 1):
ratings <- matrix(c(
  1, 2, 3, 3, 2, 1, 4, 1, 2, NA, NA, NA,
  1, 2, 3, 3, 2, 2, 4, 1, 2, 5,  NA, 3,
  NA, 3, 3, 3, 2, 3, 4, 2, 2, 5,  1,  NA,
  1, 2, 3, 3, 2, 4, 4, 1, 2, 5,  1,  NA
), nrow = 12, ncol = 4)
krippendorff_alpha(ratings, level = "nominal")

# 2. Interval ratings:
set.seed(113)
r1 <- rnorm(30, 0, 1)
r2 <- r1 + rnorm(30, 0, 0.3)
krippendorff_alpha(cbind(r1, r2), level = "interval")
```

```
# The percentile bootstrap interval for the same ratings, which
# recomputes alpha on each of B resamples of the units; the table
# gains lower_limit, upper_limit, and B_used rows. B = 200 keeps the
# example quick; a reported interval deserves the default B = 1000.
krippendorff_alpha(cbind(r1, r2), level = "interval",
                   boot = TRUE, B = 200L, seed = 113)
```

`kurtosis` *Bias-Corrected Sample Excess Kurtosis*

Description

Computes the sample excess kurtosis of a numeric vector using the bias-corrected (SAS/SPSS Type 2) formula. Excess kurtosis measures tailedness relative to the normal distribution: zero matches a normal, positive values indicate heavier tails ("leptokurtic"), negative values indicate lighter tails ("platykurtic").

Usage

```
kurtosis(x, na_rm = TRUE)
```

Arguments

`x`	A numeric vector.
`na_rm`	Logical. If `TRUE` (the default), missing values are removed before computation. If `FALSE`, the result is `NA` when `x` contains any `NA`.

Details

The reported value is

$$\hat{\gamma}_2^{(2)} = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s} \right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)},$$

where s is the (divisor-$n-1$) sample standard deviation. Subtracting the asymptotic correction $3(n-1)^2/((n-2)(n-3))$ centers the statistic at 0 for a normal distribution; that is, *excess* kurtosis is reported (rather than "raw" kurtosis, which centers at 3).

Why isn't this in base R? See the same Details in `skewness`: R Core defers higher-order moment statistics to contributed packages, partly because multiple formulas (biased Type 1, bias-corrected Type 2, Minitab Type 3) coexist. DMAR adopts Type 2, the form most common in psychometric reporting and used internally by `descriptives`.

Diagnostic interpretation. As a rough rule of thumb, $|\text{kurtosis}| > 7$ is sometimes flagged as indicative of departures from normality large enough to threaten normal-theory inference (e.g., maximum likelihood estimation in factor analysis or structural equation modeling).

Value

A single numeric value: the bias-corrected sample excess kurtosis, or `NA_real_` when fewer than four non-missing observations are available or when the sample standard deviation is zero.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Joanes, D. N., & Gill, C. A. (1998). Comparing measures of sample skewness and kurtosis. *The Statistician, 47*(1), 183–189. doi:10.1111/14679884.00122

See Also

`skewness`, `descriptives`

Other descriptive statistics: `descriptives()`, `skewness()`

Examples

```
# Normal data: excess kurtosis near zero.
set.seed(113)
kurtosis(rnorm(1000))

# Heavy-tailed data: positive excess kurtosis.
kurtosis(rt(1000, df = 4))

# The classic 1:5 example: bias-corrected excess kurtosis = -1.2.
kurtosis(1:5)
```

---

limits_of_agreement *Limits of Agreement (Bland-Altman) With Confidence Intervals on the Limits*

Description

Computes the limits of agreement (LoA) of Bland and Altman (1986, 1999) between two methods of measurement applied to the same units, along with the Carkeet (2015) exact confidence intervals on the LoA themselves. The CIs treat the two limits either as a pair (the default), so that the confidence statement holds for both limits jointly, or individually, one limit at a time. Both constructions replace the approximate normal CIs originally given by Bland and Altman (1999), which are too narrow at small n. The short alias `loa()` calls the same function.

Usage

```
limits_of_agreement(
  x,
  y,
  coverage = 0.95,
  conf_level = 0.95,
  method = c("pair", "individual")
)

loa(x, y, coverage = 0.95, conf_level = 0.95, method = c("pair", "individual"))
```

Arguments

x, y	Paired numeric vectors of equal length (e.g., method A and method B applied to the same units).
coverage	Probability content of the limits of agreement. Default `0.95` (the conventional 95% LoA). The mean difference is bracketed by $\pm z_{(1+\text{coverage})/2}$ standard deviations of the differences.
conf_level	Confidence level for the CIs on the LoA themselves. Default `0.95`.
method	How the CIs on the LoA are constructed, following Carkeet (2015): `"pair"` (the default) treats the two limits as a pair, so that the confidence statement holds for both limits jointly; `"individual"` treats each limit separately. See Details for when each is appropriate.

Details

Definition. For paired observations (x_i, y_i), the Bland-Altman limits of agreement are

$$\text{LoA}_{\pm} \;=\; \bar{d} \pm k \cdot s_d,$$

where $d_i = y_i - x_i$, $\bar{d}$ is the mean of the differences, s_d is their SD, and $k = z_{(1+\text{coverage})/2}$ (for 95% coverage, $k = 1.96$). The LoA are population intervals: they describe the range within which approximately *coverage%* of *individual* differences are expected to lie if the differences are normally distributed.

CIs on the LoA themselves. The sample LoA are random variables, and Carkeet (2015) derived exact CIs for them in two forms, selected by `method`. The choice turns on what the agreement claim is about.

The pair method (the default). A Bland-Altman analysis is usually read as a statement about the range of agreement as a whole, the span from the lower to the upper LoA within which about *coverage%* of individual differences lie. That claim involves both limits at once, so the confidence statement should hold for the two limits jointly; this is the treatment Carkeet (2015) recommends for most situations. Writing $k_t(F)$ for the exact two-sided normal tolerance factor with confidence F and content equal to `coverage` (Odeh, 1978), the CI on the upper LoA is

$$\left[\, \bar{d} + k_t(\alpha/2)\, s_d,\;\; \bar{d} + k_t(1-\alpha/2)\, s_d \,\right],$$

with α equal to one minus `conf_level`, and the CI on the lower LoA is its mirror image about $\bar{d}$. The joint confidence statement runs through the probability content of the two symmetric intervals:

with confidence `conf_level`, the interval between the inner pair of bounds, $\bar{d} \pm k_t(\alpha/2)\, s_d$, captures less than *coverage%* of the population of differences, while the interval between the outer pair, $\bar{d} \pm k_t(1-\alpha/2)\, s_d$, captures more, so the pair of population limits is bracketed simultaneously. In the Bland and Altman (1986) example that Carkeet reanalyzes ($n = 17$, $\bar{d} = -2.1$, $s_d = 38.8$), the pair bounds are -2.1 ± 57.81 (inner) and -2.1 ± 119.60 (outer).

The individual method. When a single limit carries the substantive question (for example, only the upper limit matters because only differences in one direction are clinically consequential), each limit can be treated on its own. Writing $t_{p,\,n-1}(\delta)$ for the p quantile of the noncentral *t* distribution with $n-1$ degrees of freedom and noncentrality parameter $\delta = k\sqrt{n}$, the exact CI on the upper LoA is

$$\left[\, \bar{d} + \frac{s_d}{\sqrt{n}}\, t_{\alpha/2,\,n-1}(\delta),\;\; \bar{d} + \frac{s_d}{\sqrt{n}}\, t_{1-\alpha/2,\,n-1}(\delta) \,\right],$$

and the CI on the lower LoA uses $-\delta$ in place of δ. These intervals are asymmetric about the sample LoA, wider on the side away from the mean difference. In the worked example above, the individual CI on the upper LoA is $[48.9,\ 120.0]$. The confidence statement is per limit: each limit is covered with `conf_level` confidence separately, not both at once.

Numerical accuracy. The pair tolerance factors are computed by numerical integration of the Odeh (1978) chi square by normal integral, which reproduces Carkeet's Table 2 to all four printed decimals. The individual quantiles come from `stats::qt` with a noncentrality parameter, so at very large *n* their accuracy is bounded by R's noncentral *t* algorithm: near $n = 1000$ the tolerance coefficient carries an error of about 3×10^{-4}, far past the sample sizes at which the exact-versus-approximate distinction matters.

Caveats. The LoA construction assumes (i) the differences d_i are approximately normally distributed, and (ii) the difference does not systematically depend on the magnitude of the measurement (proportional bias). Both should be checked, the second by plotting d_i against $(x_i + y_i)/2$; a non-flat relationship indicates that a single set of LoA is inappropriate.

Value

A `data.frame` with rows for the mean difference, the SD of differences, the lower and upper LoA (`loa_lower`, `loa_upper`), and the lower / upper CI bounds on each LoA. The rows are the same under both methods; the construction that produced the CI bounds is recorded in the `method` attribute.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bland, J. M., & Altman, D. G. (1986). Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet, 327*(8476), 307–310.

Bland, J. M., & Altman, D. G. (1999). Measuring agreement in method comparison studies. *Statistical Methods in Medical Research, 8*(2), 135–160. doi:10.1191/096228099673819272

Carkeet, A. (2015). Exact parametric confidence intervals for Bland-Altman limits of agreement. *Optometry and Vision Science, 92*(3), e71–e80. doi:10.1097/OPX.0000000000000513

Odeh, R. E. (1978). Tables of two-sided tolerance factors for a normal distribution. *Communications in Statistics - Simulation and Computation, 7*(2), 183–201.

See Also

lin_ccc

Other agreement and measurement: R2_mixed_effects(), content_validity_index(), gwet_ac(), icc_lmer(), krippendorff_alpha(), lin_ccc(), variance_components_mls()

Examples

```
# 1. Two methods that agree well; the CIs treat the limits as a
#    pair (the default):
set.seed(113)
method_a <- rnorm(40, mean = 100, sd = 15)
method_b <- method_a + rnorm(40, mean = 0, sd = 3)
limits_of_agreement(method_a, method_b)

# 2. Each limit treated individually, for when a single limit
#    carries the substantive question:
limits_of_agreement(method_a, method_b, method = "individual")

# 3. 90% LoA with 95% CIs on the limits:
limits_of_agreement(method_a, method_b, coverage = 0.90, conf_level = 0.95)
```

---

lin_ccc *Lin's Concordance Correlation Coefficient*

Description

Computes Lin's (1989) concordance correlation coefficient (CCC) for a pair of vectors of paired observations, together with a confidence interval built on Lin's z-transformed standard error (Lin, 1989; see also the note in Lin, 2000). The CCC measures agreement (not merely correlation) between two methods of measurement: a CCC of 1 means perfect agreement ($y_i = x_i$ for all i), while Pearson's r would still be 1 for any straight-line relationship, even one with non-unit slope.

Usage

```
lin_ccc(x, y, conf_level = 0.95, method = "lin")
```

Arguments

x, y	Paired numeric vectors of equal length (e.g., the two measurement methods).
conf_level	Confidence level for the CI. Default 0.95.
method	The confidence interval method. Currently the only option is "lin", the Lin (1989) z-transformed standard error.

Details

Definition. Lin (1989) defined the CCC as

$$\rho_c \;=\; \frac{2\rho\,\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2+(\mu_x-\mu_y)^2},$$

where $\rho = \mathrm{Cor}(X,Y)$ is the Pearson correlation and the denominator is inflated by the squared mean difference and by any inequality of the two variances, so disagreement in location or scale pulls ρ_c below ρ. ρ_c factors as $\rho_c = \rho \cdot C_b$, where $C_b \in [0,1]$ is the "bias correction factor" that captures location and scale agreement, and $C_b = 1$ iff $\mu_x = \mu_y$ and $\sigma_x = \sigma_y$.

Confidence interval. The Fisher-style z-transform of the CCC, $z = \frac{1}{2}\log\{(1+\rho_c)/(1-\rho_c)\}$, has approximate variance (Lin, 1989, as corrected in Lin, 2000)

$$\mathrm{Var}(z) \;\approx\; \frac{1}{n-2}\left[\frac{(1-\rho^2)\rho_c^2}{(1-\rho_c^2)\rho^2} + \frac{2\rho_c^3(1-\rho_c)u^2}{\rho(1-\rho_c^2)^2} - \frac{\rho_c^4u^4}{2\rho^2(1-\rho_c^2)^2}\right],$$

where $u = (\mu_x-\mu_y)/\sqrt{\sigma_x\sigma_y}$. The CI is built on the z-scale and back-transformed via $\tanh$.

This variance is derived under bivariate normality, and the interval inherits that assumption. Under normality its coverage is modestly below the nominal rate in small samples (roughly 0.92 to 0.94 at n of 10 to 20 for a nominal 0.95) and approaches the nominal rate as n grows (about 0.94 at $n = 50$ for a moderate CCC). With clearly skewed data the situation is worse and more data do not repair it: with heavy-tailed or log-normal style measurements the interval can cover far below the nominal rate at any sample size (Carrasco, Jover, King, & Chinchilli, 2007). With such data, transform toward symmetry before computing the CCC, or use a bootstrap interval on $\hat{\rho}_c$.

Value

A `data.frame` with rows for the CCC point estimate, the lower and upper CI limits, and decomposition components (Pearson r, accuracy C_b, location-shift u, scale-shift v).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Carrasco, J. L., Jover, L., King, T. S., & Chinchilli, V. M. (2007). Comparison of concordance correlation coefficient estimating approaches with skewed data. *Journal of Biopharmaceutical Statistics, 17*(4), 673–684. doi:10.1080/10543400701329463

Lin, L. I.-K. (1989). A concordance correlation coefficient to evaluate reproducibility. *Biometrics, 45*(1), 255–268.

Lin, L. I.-K. (2000). A note on the concordance correlation coefficient. *Biometrics, 56*(1), 324–325. doi:10.1111/j.0006341X.2000.00324.x

See Also

`limits_of_agreement`

Other agreement and measurement: `R2_mixed_effects()`, `content_validity_index()`, `gwet_ac()`, `icc_lmer()`, `krippendorff_alpha()`, `limits_of_agreement()`, `variance_components_mls()`

Examples

```
# 1. Two methods of measuring the same quantity:
set.seed(113)
method_a <- rnorm(40, mean = 100, sd = 15)
method_b <- method_a + rnorm(40, mean = 2, sd = 5)
lin_ccc(method_a, method_b)

# 2. Compare CCC with Pearson r when there is a systematic offset:
lin_ccc(method_a, method_a + 5)$value[1:2]   # CCC < r
cor(method_a, method_a + 5)                  # Pearson r = 1
```

---

manova_split_plot *Mixed-Design Multivariate ANOVA With All Four Test Statistics*

Description

Computes the multivariate analysis of variance for a mixed-design (one between-subjects factor and one within-subjects factor), returning Wilks's Λ, Pillai's trace, Hotelling-Lawley trace, and Roy's largest root, each with the associated F-approximation, degrees of freedom, and p-value. The three effects, between-subjects (A), within-subjects (B), and the interaction ($A \times B$), are tested separately. Wraps `Anova` and returns the result in the tidy DMAR style.

Usage

```
manova_split_plot(data, within, between, ss_type = 3L)
```

Arguments

`data`	A `data.frame` in wide format, one row per subject, with the within-subjects measurements in separate columns and the between-subjects factor as one additional column.
`within`	Character vector of column names holding the repeated measures values (one column per level of the within- subjects factor). Must be in the canonical level order.
`between`	Character name of the between-subjects factor column in `data`.
`ss_type`	The sum-of-squares type for the between-subjects effects, passed through to `Anova`. Accepts the integers `1`, `2`, or `3` or the equivalent Roman- numeral strings `"I"`, `"II"`, or `"III"`, and defaults to `3L` (Type III). The chosen type is recorded in the returned table (see **Value**). Type I (`1` or `"I"`) is not available for this mixed design, because `Anova` computes only Type II and Type III sums of squares for the multivariate repeated measures path; requesting it raises an error.

Details

The four statistics. For an effect with H and E hypothesis- and error-cross-products matrices:

- Wilks's $\Lambda = \det(E)/\det(E+H)$
- Pillai's trace $V = \mathrm{tr}(H(E+H)^{-1})$
- Hotelling-Lawley trace $T_0^2 = \mathrm{tr}(HE^{-1})$
- Roy's largest root $\theta = \lambda_1(HE^{-1})$

When to use which. Pillai's trace is the most robust to departures from the multivariate normal / homogeneous-covariance assumptions. Wilks's Λ is the most widely reported. Roy's largest root is the most powerful when the alternative concentrates on a single dimension. The four statistics agree exactly when the effect has 1 numerator degree of freedom.

Sum-of-squares type. The between-subjects effects are computed by `Anova` using the sum-of-squares type selected through `ss_type` (Type III by default). Type II conditions each effect on the others that do not contain it, and Type III conditions each effect on every other effect in the model; for a single between-subjects factor the two coincide, and they can differ once additional between-subjects terms are present. Type I, the sequential decomposition, is not available here: `Anova` computes only Type II and Type III for the multivariate repeated measures path. The type in force is reported in the returned table so the analysis is self-documenting.

Dependency. Requires the **car** package on CRAN.

Value

A `data.frame` with rows for each of the three effects crossed with each of the four multivariate statistics, plus one trailing row recording the sum-of-squares type. Columns: `effect`, `statistic_name`, `statistic_value`, `F_approx`, `df_1`, `df_2`, `p_value`. The final row has `effect == "sum_of_squares_type"` and carries the chosen type (`1`, `2`, or `3`) in its numeric `statistic_value`; its remaining numeric columns are `NA`, so the `statistic_value` column stays numeric.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 7 on higher-order designs and Chapter 14.)

Rencher, A. C., & Christensen, W. F. (2012). *Methods of multivariate analysis* (4th ed.). Wiley.

See Also

`Anova`, `manova`, `anova_within_two_way`

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `correlations_test()`, `equivalence_r()`, `equivalence_smd()`, `factorial_anova()`, `mauchly_test()`, `mixed_anova()`, `obrien_test()`, `pairwise_within()`, `randomization_test()`, `randomization_test_paired()`, `regions_of_significance()`, `simple_effects_AB()`, `summary_t_test()`, `welch_t()`

Other mixed models: R2_mixed_effects(), R2_mixed_effects_decomposition(), icc_lmer(), mixed_anova(), ss_aipe_mixed_effects(), ss_aipe_mixed_effects_sensitivity(), ss_power_mixed_effects(), ss_power_split_plot_anova()

Examples

```
# Two groups of ten measured at three times. Both groups start at the
# same place and rise over time, and group B rises twice as fast, so the
# data generating means differ in slope as well as in level.
set.seed(113)
n_per <- 10
d <- data.frame(
  subject = factor(1:(2 * n_per)),
  group   = factor(rep(c("A", "B"), each = n_per)),
  t1      = c(rnorm(n_per, 0,   1), rnorm(n_per, 0,   1)),
  t2      = c(rnorm(n_per, 0.4, 1), rnorm(n_per, 0.8, 1)),
  t3      = c(rnorm(n_per, 0.8, 1), rnorm(n_per, 1.6, 1))
)

# Rows are the between-subjects effect (labeled A), the within-subjects
# effect (labeled B), and their interaction, each with all four
# multivariate criteria, followed by a row recording the sum-of-squares
# type. The multivariate tests make no sphericity assumption, which is
# what recommends them over the univariate repeated measures F and its
# epsilon corrections.
manova_split_plot(d, within = c("t1", "t2", "t3"), between = "group")
```

---

mauchly_test *Mauchly's Test of Sphericity for a One-Way Within-Subjects Design*

Description

Tests the null hypothesis that the covariance matrix of the orthonormal contrasts among the k repeated measurements is proportional to the identity (the *sphericity* assumption underlying univariate repeated measures F-tests).

Usage

```
mauchly_test(x, id = NULL, time = NULL, outcome = NULL)
```

Arguments

x	Either an $n \times k$ numeric matrix or data.frame (rows = subjects, columns = repeated measurements); *or* a long-format data.frame together with id, time, and outcome column names.
id	Column name in x identifying the subject when x is in long format (NULL otherwise).

`time` Column name in `x` identifying the within-subjects factor level when `x` is in long format (`NULL` otherwise).

`outcome` Column name in `x` identifying the dependent variable when `x` is in long format (`NULL` otherwise).

Details

Sphericity is the assumption that the variances of all pairwise differences among the k levels are equal, equivalently, that the covariance matrix Σ_C of any orthonormal set of $k-1$ contrasts among the levels is proportional to the identity. Mauchly's (1940) test statistic is

$$W = \frac{\det(\hat{\Sigma}_C)}{\left(\operatorname{tr}(\hat{\Sigma}_C)/(k-1)\right)^{k-1}},$$

and the chi square approximation

$$X^2 = -\,m\,\log W \quad \text{with } m = (n-1) - \frac{2(k-1)^2 + (k-1) + 2}{6\,(k-1)}$$

has approximately $(k-1)k/2 - 1$ degrees of freedom under H_0. The reported p-value uses Box's (1949) second-order correction, a weighted combination of the chi square tails on $(k-1)k/2 - 1$ and $(k-1)k/2 + 3$ degrees of freedom, which improves the first-order approximation in small samples; this matches `mauchly.test`.

When sphericity is rejected, the univariate F test is liberal; correct using the Greenhouse-Geisser, Huynh-Feldt, or lower-bound epsilon adjustments via `epsilon_corrections` or directly via `anova_within`.

The test is only defined for $k \geq 3$; with $k = 2$, sphericity is trivially true and the function returns `W = 1, p = 1`.

Value

A one-row `data.frame` with columns `W` (Mauchly's statistic), `statistic` (the chi square approximation), `df`, `p_value`, `n_subjects`, `n_levels`, and `method`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Mauchly, J. W. (1940). Significance test for sphericity of a normal n-variate distribution. *Annals of Mathematical Statistics, 11*(2), 204–209.

Box, G. E. P. (1949). A general distribution theory for a class of likelihood criteria. *Biometrika, 36*(3/4), 317–346.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 11 for sphericity in within-subjects designs.)

See Also

epsilon_corrections, anova_within

Other within-subjects analysis: anova_within(), anova_within_two_way(), epsilon_corrections(), pairwise_within(), plot_trajectories_fitted()

Other hypothesis tests: adjusted_means(), ancova(), anova_within(), ci_dunnett(), ci_scheffe(), ci_tukey_kramer(), compare_cov_structures(), contrast_test(), correlations_test(), equivalence_r(), equivalence_smd(), factorial_anova(), manova_split_plot(), mixed_anova(), obrien_test(), pairwise_within(), randomization_test(), randomization_test_paired(), regions_of_significance(), simple_effects_AB(), summary_t_test(), welch_t()

Examples

```
# Wide-format example: simulated within-subjects data with 4 levels.
set.seed(113)
Y <- matrix(rnorm(20 * 4), nrow = 20)
mauchly_test(Y)

# Long-format example using built-in nlme::Orthodont (4 ages per subject).
mauchly_test(nlme::Orthodont, id = "Subject", time = "age",
             outcome = "distance")
```

---

measurement_alignment *Approximate Measurement Invariance by Factor Alignment*

Description

Estimates the group factor means and factor variances that make the measurement parameters as nearly invariant as possible across groups, following the alignment method of Asparouhov and Muthén (2014). The invariance ladder in measurement_invariance asks whether loadings and intercepts are exactly equal across groups. With many groups that hypothesis is essentially never true, the ladder stalls at configural invariance, and the comparison of factor means the researcher wanted never happens. Alignment takes the configural solution as given and searches for the group factor means and variances that concentrate the noninvariance in a few parameters instead of spreading it thinly over many, so that factor means stay comparable without claiming that exact invariance holds. Requires **lavaan**.

Usage

```
measurement_alignment(
  data,
  items,
  group,
  model = NULL,
  alignment = c("fixed", "free"),
  estimator = "ML",
  n_starts = 10,
```

```
  seed = NULL,
  epsilon = 0.01
)
```

Arguments

data	A data.frame holding the items and the grouping variable.
items	Character vector naming the indicator columns of data. Three or more items are required, and each must be numeric.
group	Single character string naming the grouping column of data. Two or more groups are required.
model	Optional **lavaan** model syntax for the measurement model, for the case where the default single factor over items is not what is wanted (a residual covariance, for example). The syntax must define exactly one latent variable and its indicators must be exactly items. The factor is standardized (mean 0, variance 1) in every group by this function, so do not set its scale in the syntax. When NULL (the default) the model f =~ item1 + item2 + ... is used.
alignment	Identification rule for the metric of the factor, either "fixed" (the default) or "free". Under "fixed" the first group's factor mean is held at 0 and its factor variance at 1. Under "free" the group factor means are constrained to average 0 and the group factor standard deviations to have a geometric mean of 1, so no group serves as the reference. The choice sets the metric the factor means and variances are reported on; see Details.
estimator	Estimator passed to **lavaan** for the configural model. Defaults to "ML"; "MLR" supplies the robust corrections.
n_starts	Number of starting values for the optimizer, a positive integer (default 10). The first start sets every factor mean to 0 and every factor variance to 1, the second uses a median based heuristic, and any remaining starts are random. The simplicity function has local minima, so several starts is the default rather than a refinement.
seed	Optional integer seed for the random starts. Defaults to NULL, which uses the caller's random number state and leaves it alone. When an integer is supplied the seed is set locally and the caller's state is restored on exit.
epsilon	Smoothing constant $\epsilon > 0$ inside the component loss function, default 0.01. Smaller values push the loss closer to $\sqrt{\lvert x \rvert}$ and make the surface harder to optimize; larger values smooth it at the cost of blurring the distinction between a few large differences and many small ones.

Details

Alignment starts from the *configural* solution: the multiple group single factor model with every loading, intercept, and residual variance free across groups and the factor standardized (mean 0 and variance 1) in each group. Write λ^0_{gi} and ν^0_{gi} for the resulting loading and intercept of item i in group g. That solution is not unique: for any group factor means α_g and factor variances ψ_g the reparameterized measurement parameters

$$\lambda_{gi} = \lambda^0_{gi} / \sqrt{\psi_g}$$

$$\nu_{gi} = \nu^0_{gi} - \alpha_g \lambda^0_{gi} / \sqrt{\psi_g}$$

reproduce the observed means and covariances exactly and so fit the data identically. Alignment picks the member of that family whose measurement parameters are closest to invariant, by minimizing the total simplicity function

$$F = \sum_i \sum_{g_1 < g_2} w_{g_1 g_2} f(\lambda_{g_1 i} - \lambda_{g_2 i}) + \sum_i \sum_{g_1 < g_2} w_{g_1 g_2} f(\nu_{g_1 i} - \nu_{g_2 i})$$

over α_g and ψ_g, with weights $w_{g_1 g_2} = \sqrt{N_{g_1} N_{g_2}}$ and the component loss function

$$f(x) = \sqrt{\sqrt{x^2 + \epsilon}} = (x^2 + \epsilon)^{1/4}.$$

The fourth root is the substance of the method, not a technical detail. A squared loss would spread a fixed amount of noninvariance evenly over all the parameters, because halving two differences beats zeroing one. The fourth root is concave in $|x|$, so the marginal penalty falls as a difference grows: the criterion prefers a solution in which most parameters agree closely and a few disagree substantially, which is exactly the pattern of approximate invariance a researcher wants to find and report. The constant ϵ only rounds off the kink of $\sqrt{|x|}$ at zero so the criterion is differentiable. One consequence is worth knowing: each of the $2I$ terms in a group pair contributes at least $\epsilon^{1/4}$, so the smallest attainable value of F is $2I\epsilon^{1/4} \sum_{g_1 < g_2} w_{g_1 g_2}$, reached only when every aligned parameter is exactly invariant. The achieved value is comparable across runs on the same items, groups, and ϵ, not across data sets.

Two constraints identify the solution. The scale of the factor is genuinely undetermined by F: multiplying every $\sqrt{\psi_g}$ and every α_g by the same constant leaves the aligned intercepts alone and shrinks the aligned loadings toward each other, so without a constraint the criterion is minimized by letting the factor variances run away. The location constraint plays the same role in the limiting case of exact invariance, where shifting all the factor means by a constant leaves F unchanged. Under `alignment = "fixed"` the constraints are $\alpha_1 = 0$ and $\psi_1 = 1$. Under `alignment = "free"` they are $\sum_g \alpha_g = 0$ and $\prod_g \sqrt{\psi_g} = 1$, which treats the groups symmetrically and is the more natural choice when no group is a meaningful reference. The two rules give genuinely different solutions, not a relabeling of one another, because F is not invariant to shifting the factor means.

The rule also sets the metric the answer is reported on, which matters when the estimates are compared with anything else. The constraint $\psi_1 = 1$ makes the first group's factor the common metric, so a reported factor mean of 0.4 says that group's factor mean is 0.4 of the *first group's* factor standard deviations above the first group's, and a reported factor variance of 1.3 says its factor variance is 1.3 times the first group's. Under `"free"` the common metric is the one in which the group factor standard deviations have a geometric mean of 1, and the factor means are deviations from their own average on that metric. Simulating data with known group factor means and then comparing them with the estimates requires dividing the generating means by the generating factor standard deviation of the reference group first.

The minimization runs `optim` with the BFGS method and the analytic gradient, over the factor means α_g and the log factor standard deviations $\log \sqrt{\psi_g}$, reduced to the coordinates the identification rule leaves free. The optimizer is run from `n_starts` starting values and the best solution is kept, because a single start is not safe. Two things can go wrong, and they are different problems.

The first is ordinary multimodality. When several loadings and intercepts are noninvariant the criterion has more than one local minimum, typically within a percent or so of each other in value but at different factor means. The number of distinct minima the starts reached is returned in the

"n_optima" attribute and the value achieved from each start in "simplicity_starts". More than one is a signal to raise n_starts and to check whether the reported solution is stable.

The second is a degenerate branch, and it is the one that bites. Sending the factor variances of every group but the reference off to infinity drives those groups' aligned loadings to zero, which flattens the loading half of the criterion; the optimizer stops there and reports convergence with factor variances of 10^{36} or Inf. That is not an estimate, and on real data a meaningful share of random starts finds it. A start whose factor standard deviations or factor means leave a very wide sanity range (a factor of 10^4 either way) is therefore discarded and its entry in "simplicity_starts" is Inf. If every start is discarded the function stops rather than return the degenerate solution.

Item level invariance is summarized with the R^2 measure of Asparouhov and Muthén (2014), which asks how much of the group to group variation in an item's configural parameter is accounted for by the estimated factor means and variances alone. Let $\bar{\lambda}_i$ and $\bar{\nu}_i$ be the aligned parameters of item i averaged over groups. The factor means and variances by themselves imply the configural values $\sqrt{\psi_g}\bar{\lambda}_i$ and $\bar{\nu}_i + \alpha_g\bar{\lambda}_i$. Write T_i for the sum over groups of the squared configural parameter of item i and E_i for the sum over groups of its squared departure from that implied value. Then

$$R_i^2 = 1 - E_i/T_i,$$

computed separately for the loadings and for the intercepts. A value of 1 for every item on the loadings is metric invariance, and on both loadings and intercepts is scalar invariance. The complementary view is the "item_loss" attribute, which splits the achieved simplicity function into the contribution of each item, so the items carrying the noninvariance can be named.

The criterion adds up raw parameter differences, so items on very different measurement scales do not contribute equally: an item whose raw variance is a hundred times another's dominates the sum. When the items are not already on a common metric, put them on one (standardizing them is the simple choice) before aligning.

The method is defined for two groups and is computed here for two, but it has little to offer there. With two groups the standard invariance ladder is tractable and interpretable, and the alignment criterion has only one group pair to work with. Alignment earns its keep when the number of groups makes exact invariance implausible and the ladder uninformative.

Value

A data.frame (class dmar_tbl) with one row per group and columns group (the group label), n (the number of cases the configural model used in that group), factor_mean (the estimated α_g), and factor_variance (the estimated ψ_g). The rows follow the group ordering **lavaan** uses.

Attributes carry the rest of the solution:

aligned_loadings, aligned_intercepts Matrices of the aligned λ_{gi} and ν_{gi}, groups in rows and items in columns.

configural_loadings, configural_intercepts Matrices of the configural λ^0_{gi} and ν^0_{gi}, same layout.

simplicity_function The achieved minimum of F.

simplicity_starts The value of F achieved from each starting value, in start order. A start the optimizer could not complete, or one that ended in the degenerate branch described in Details, is recorded as Inf and discarded.

converged TRUE when the optimizer reported convergence at the retained solution.

`n_starts` The number of starting values used.

`n_optima` The number of distinct local minima the converged starts reached.

`alignment` The identification rule used, `"fixed"` or `"free"`.

`epsilon` The smoothing constant used.

`R2_loadings`, `R2_intercepts` Named numeric vectors, one entry per item, holding the per item R^2 invariance measures defined in Details.

`R2_total` The same two measures pooled over items, as a named numeric vector with elements `loadings` and `intercepts`. This is the overall effect size of approximate invariance reported by Asparouhov and Muthén (2014).

`item_loss` Matrix with one row per item and columns `loadings`, `intercepts`, and `total`, splitting the achieved simplicity function into per item contributions.

`fit` The configural **lavaan** fit object.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Asparouhov, T., & Muthén, B. (2014). Multiple-group factor analysis alignment. *Structural Equation Modeling, 21*(4), 495–508. doi:10.1080/10705511.2014.919210

Marsh, H. W., Guo, J., Parker, P. D., Nagengast, B., Asparouhov, T., Muthén, B., & Dicke, T. (2018). What to do when scalar invariance fails: The extended alignment method for multi-group factor analysis comparison of latent means across many groups. *Psychological Methods, 23*(3), 524–545. doi:10.1037/met0000113

Muthén, B., & Asparouhov, T. (2014). IRT studies of many groups: The alignment method. *Frontiers in Psychology, 5*, Article 978. doi:10.3389/fpsyg.2014.00978

Muthén, B., & Asparouhov, T. (2018). Recent methods for the study of measurement invariance with many groups: Alignment and random effects. *Sociological Methods & Research, 47*(4), 637–664. doi:10.1177/0049124117701488

Robitzsch, A. (2025). *sirt: Supplementary item response theory models*. R package version 4.2-133. https://CRAN.R-project.org/package=sirt

See Also

`measurement_invariance` for the exact invariance ladder alignment is meant to rescue; `cfa_1` for the single group measurement model; `htmt` for discriminant validity of the same items.

Other multivariate and latent variable methods: `average_variance_extracted()`, `bifactor_indices()`, `cfa_1()`, `cfa_2()`, `cfa_k()`, `ci_eigenvalue()`, `common_method_marker()`, `common_method_single_factor()`, `dmacs()`, `ecvi()`, `htmt()`, `irt_grm()`, `irt_information()`, `measurement_invariance()`, `procrustes_phi()`, `simple_structure()`

Examples

```
# Five groups that differ in factor mean and factor variance, with two
# deliberately noninvariant measurement parameters (the loading of item
# 3 in group 2 and the intercept of item 5 in group 4). The first group's
# factor standard deviation is 1, which is the metric the default "fixed"
# rule reports on, so the estimates compare directly with the generating
# values below.
set.seed(113)
n_g <- c(400, 450, 380, 500, 420)
factor_mean <- c(0, 0.30, -0.50, 0.80, 0.20)
factor_sd   <- c(1, 1.20,  0.80, 1.10, 0.90)
Lambda <- matrix(0.8, nrow = 5, ncol = 6)
Nu     <- matrix(1.0, nrow = 5, ncol = 6)
Lambda[2, 3] <- 0.3
Nu[4, 5]     <- 1.8
d <- do.call(rbind, lapply(1:5, function(g) {
  eta <- rnorm(n_g[g], factor_mean[g], factor_sd[g])
  x <- sapply(1:6, function(i)
    Nu[g, i] + Lambda[g, i] * eta + rnorm(n_g[g], 0, 0.6))
  data.frame(x, cohort = paste0("cohort_", g))
}))
names(d)[1:6] <- paste0("x", 1:6)

out <- measurement_alignment(d, items = paste0("x", 1:6),
                             group = "cohort", seed = 113)
out                                  # recovered means and variances
attr(out, "R2_loadings")             # item 3 stands out
attr(out, "R2_intercepts")           # item 5 stands out
attr(out, "item_loss")

# Holzinger and Swineford's verbal tests across four groups formed by
# crossing school with sex. The raw tests are on very different scales,
# so they are standardized first (see Details).
data(holzinger_swineford)
hs <- holzinger_swineford
hs$school_sex <- interaction(hs$school, hs$sex, sep = ", ")
verbal <- c("t5_general_information", "t6_paragraph_comprehension",
            "t7_sentence", "t8_word_classification", "t9_word_meaning")
hs[verbal] <- scale(hs[verbal])
ma <- measurement_alignment(hs, items = verbal, group = "school_sex",
                            seed = 113)
ma

# The broom verbs: one row per group, and the alignment summary.
generics::tidy(ma)
generics::glance(ma)
```

`measurement_invariance`

Measurement Invariance Across Groups

Description

Fits the standard ladder of multiple group invariance models for a measurement model of any number of factors and reports the comparison table researchers actually use: configural invariance (same pattern, all parameters free by group), metric (equal loadings; required before comparing relations involving the factor), scalar (equal loadings and intercepts; required before comparing factor means), and strict (equal residual variances as well; required before comparing observed-score variances). With ordered indicators a thresholds rung comes first, because thresholds rather than intercepts carry the location information (Wu & Estabrook, 2016); with dichotomous indicators the two are not separately identified, so that rung is folded into metric. Each rung is tested against the previous with a likelihood ratio test, scaled when the estimator in force carries a robust test, which is what declaring ordered items arranges, and the practical-fit changes (delta CFI, delta RMSEA) are reported alongside, since with large samples the chi square will flag trivial differences (Cheung & Rensvold, 2002, suggest delta CFI of about -.01 as a red flag). The fitted **lavaan** objects come back as an attribute, so score tests and partial invariance refits do not require refitting the ladder. Requires **lavaan**.

Usage

```
measurement_invariance(
  data,
  model = NULL,
  group,
  items = NULL,
  levels = NULL,
  ordered = NULL,
  estimator = "ML",
  missing = "listwise",
  group_partial = NULL,
  parameterization = c("delta", "theta"),
  ...
)
```

Arguments

`data`	A `data.frame` with the items and the grouping variable.
`model`	The measurement model, given either as **lavaan** model syntax (a single string, or a character vector of lines that is collapsed with newlines, such as `c("visual =~ x1 + x2 + x3", "verbal =~ x4 + x5 + x6")`) or as a named list mapping each factor name to a character vector of its item names, from which the syntax is built. Exactly one of `model` and `items` must be supplied.
`group`	Single character string naming the grouping column (two or more groups).
`items`	Character vector (three or more) naming the indicator columns of a single factor. A convenience for the one-factor (congeneric) case of `cfa_1`, equivalent to `model = "f =~ item_1 + item_2 + ..."`. Exactly one of `model` and `items` must be supplied.
`levels`	Which rungs of the ladder to fit, in order: any leading subset of the ladder in force. With continuous indicators the ladder is `c("configural", "metric",`

`"scalar"`, `"strict"`); with ordered indicators it is `c("configural", "thresholds", "metric","scalar", "strict")`; when every indicator is ordered and dichotomous it is `c("configural", "metric", "scalar")`, because neither the thresholds nor the strict rung is identified for a two-category item (see `ordered` below). `NULL` (the default) fits the whole ladder in force.

`ordered` Ordered categorical (including binary) items: `TRUE` for every indicator in the model, or a character vector naming the ordered ones. `NULL` (default) treats all indicators as continuous. Declaring ordered items switches the ladder and, unless the estimator already carries the mean and variance adjusted test, switches the estimator (see `estimator` below). The number of observed categories per declared item is counted before the ladder is built, because a dichotomous item has a single threshold that is not separately identified from the intercept of its underlying response (Millsap & Yun-Tein, 2004; Wu & Estabrook, 2016), and its residual variance is not a free parameter under either parameterization. When every declared item is dichotomous the thresholds rung is folded into metric, and the strict rung is dropped as well unless a continuous indicator is left to carry a residual variance; when only some declared items are dichotomous, both rungs are kept and a message names those items, whose thresholds and residual variances the rungs leave untested.

`estimator` Estimator passed to **lavaan**. Defaults to `"ML"`; use `"MLR"` for robust corrections, in which case the likelihood ratio tests use the scaled difference. When `ordered` is supplied, a maximum likelihood or generalized least squares (the `"GLS"` label; Browne, 1974) estimator is replaced by `"WLSMV"`, and `"DWLS"` and `"ULS"` are replaced by `"WLSMV"` and `"ULSMV"`. That second substitution changes only the test: in **lavaan** `"WLSMV"` is `"DWLS"` and `"ULSMV"` is `"ULS"` with the mean and variance adjusted statistic, which is what makes the rung-to-rung difference test asymptotically valid for ordered data. The one categorical estimator left alone is `"WLS"`, the full weight matrix estimator of Browne (1984) and Muthén (1984), whose statistic is asymptotically chi square already and whose weight matrix a substitution would silently change.

`missing` Missing data handling passed to **lavaan**: one of `"listwise"` (the default), `"ml"` or `"fiml"` (full information maximum likelihood, the reason to reach for this argument with continuous items and incomplete cases), or `"pairwise"`. Full information maximum likelihood is not available with the categorical estimator; asking for both is an error.

`group_partial` Character vector of parameters to leave free across groups at every rung, passed to **lavaan**'s `group.partial`. Either user-supplied parameter labels or **lavaan** parameter specifications such as `"visual =~ x2"` or `"x2 ~1"`. This is the partial invariance case of Byrne, Shavelson, and Muthén (1989). `NULL` (default) constrains everything the rung calls for.

`parameterization`
Identification of ordered indicators, passed to **lavaan**: `"delta"` (the default, lavaan's) or `"theta"`. Residual variances are free parameters only under `"theta"`, so a strict rung requested with ordered items switches to `"theta"` with a message. Ignored when no item is ordered.

`...` Further arguments passed to `cfa` at every rung (for example `std.lv`, `orthogonal`, `cluster`).

Details

The models are nested by construction, each adding equality constraints across groups to the previous. With continuous indicators the constraint sets are none (beyond the configuration), then `group.equal = "loadings"`, then `c("loadings", "intercepts")`, then `c("loadings", "intercepts", "residuals")`.

The ordered ladder differs, and the difference is substantive rather than cosmetic. For an ordered indicator the observed response is a coarsening of an underlying continuous response at a set of thresholds, and it is the thresholds, not an intercept, that locate the item on the latent scale. Constraining loadings while leaving thresholds free across groups therefore does not deliver what metric invariance is supposed to deliver, and the accepted sequence (Millsap & Yun-Tein, 2004; Wu & Estabrook, 2016) constrains thresholds first: `"thresholds"`, then `c("thresholds", "loadings")` for metric, then `c("thresholds", "loadings", "intercepts")` for scalar, then adding `"residuals"` for strict. The intercept rung is not vacuous even when every indicator is ordered: once thresholds and loadings are constrained, **lavaan** frees the underlying-response intercepts in the non-reference groups, and the scalar rung is what returns them to zero and lets the latent means be estimated instead. Residual variances, however, are free parameters only under the theta parameterization, so `parameterization` switches to `"theta"` when a strict rung is requested with ordered items.

Dichotomous items are the exception the ordered ladder has to make room for. A two-category item contributes one threshold, and that threshold, the intercept of the underlying response, and its residual variance are not separately identified: the data give one proportion per group per item, which pins down a single standardized location and nothing else (Millsap & Yun-Tein, 2004; Wu & Estabrook, 2016). Constraining thresholds alone across groups is then a reparameterization rather than a restriction, since **lavaan** frees the underlying-response intercepts by exactly as many parameters as the constraint removes, and the residual variances stay fixed at one in every group whichever parameterization is in force. So the function counts the observed categories of each declared ordered indicator before building the ladder, and a message explains what the count implies. When all of them are dichotomous the thresholds rung is folded into metric, which constrains thresholds and loadings together; the strict rung goes too when every indicator in the model was declared ordered, leaving configural, metric, scalar. A continuous indicator alongside dichotomous ones keeps the strict rung, since its residual variance is a free parameter. When only some declared items are dichotomous the full ordered ladder is kept, since the polytomous items still carry testable thresholds and residual variances, and the message names the dichotomous items so their contribution to those two rungs is not overread. Every rung that is dropped is one that would have cost zero degrees of freedom.

When the estimator carries a robust test, the difference between two chi square statistics is not itself chi square distributed. `lavTestLRT` then returns the scaled difference test (the Satorra-Bentler or Satorra correction, chosen by **lavaan** to match the test in force), and that is what `delta_chi_square` and `p_value` report; the `"test"` attribute names the test used. In that case the model-level `chi_square`, `p_chi_square`, `cfi`, and `rmsea` columns are lavaan's scaled and robust versions, so `delta_chi_square` will not equal the difference of consecutive `chi_square` entries. That is a property of scaled difference testing, not an inconsistency.

Which estimators carry such a test is worth being precise about, because a naive chi square difference on ordered data is not asymptotically valid. `"MLM"`, `"MLMV"`, `"MLR"`, `"WLSM"`, `"WLSMV"`, `"ULSM"`, and `"ULSMV"` carry one. `"DWLS"` and `"ULS"` do not, which is why declaring ordered items promotes them to `"WLSMV"` and `"ULSMV"`: the discrepancy function and hence the parameter estimates are untouched, and only the statistic changes. The remaining case is `"WLS"`, the full weight matrix esti-

mator, whose statistic is asymptotically chi square under the theory of Browne (1984) and Muthén (1984) and so needs no correction; the `"test"` attribute reports the standard difference test there. The sample size that theory asks for is large, which is why full weighted least squares is rarely the right choice in practice, but that is a separate matter from whether the difference test is the right one.

A rung whose constraints cost no degrees of freedom tests nothing, so `delta_chi_square` and `p_value` are `NA` when `delta_df` is zero rather than reporting a difference in chi square that is numerical noise and can come out negative. The situation is reported in a message. It arises with dichotomous items (the ladder above avoids the two rungs where it is structural) and, for instance, with `group_partial` specifications that free every parameter a rung would constrain.

Failure at a rung does not end the conversation: partial invariance (freeing the offending parameter) is the usual next step, for which `group_partial` refits the whole ladder with named parameters free, and the `"fits"` attribute gives the fitted objects to `lavTestScore` for a score test of which constraint is doing the damage. This function deliberately reports the standard ladder rather than automating modification searches.

Value

A tidy wide `data.frame` (class `dmar_tbl`) with one row per fitted level and columns `level` (label), `chi_square`, `df`, `p_chi_square` (exact-fit test), `cfi`, `rmsea`, and, from the second row on, the step-comparison columns `delta_chi_square`, `delta_df`, `p_value` (the likelihood ratio test against the previous rung), `delta_cfi`, and `delta_rmsea`. `delta_chi_square` and `p_value` are `NA` wherever `delta_df` is zero, since a difference test on zero degrees of freedom tests nothing. Attributes carry the non-numeric information: `"fits"`, the named list of fitted **lavaan** objects, one per level; `"estimator"`, the estimator actually used; `"ordered"`, `TRUE` when any indicator was declared ordered; `"test"`, naming the chi square difference test used (`NA` when a single rung was fit and nothing was compared); `"fit_indices"`, `"standard"` or `"robust"` according to which version of the fit indices is tabled; and `"model"`, the **lavaan** syntax that was fitted.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Browne, M. W. (1974). Generalized least squares estimators in the analysis of covariance structures. *South African Statistical Journal, 8*, 1–24.

Browne, M. W. (1984). Asymptotically distribution-free methods for the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology, 37*(1), 62–83.

Byrne, B. M., Shavelson, R. J., & Muthén, B. (1989). Testing for the equivalence of factor covariance and mean structures: The issue of partial measurement invariance. *Psychological Bulletin, 105*(3), 456–466.

Cheung, G. W., & Rensvold, R. B. (2002). Evaluating goodness-of-fit indexes for testing measurement invariance. *Structural Equation Modeling, 9*(2), 233–255. doi:10.1207/S15328007SEM0902_5

Meredith, W. (1993). Measurement invariance, factor analysis and factorial invariance. *Psychometrika, 58*(4), 525–543. doi:10.1007/BF02294825

Millsap, R. E. (2011). *Statistical approaches to measurement invariance*. Routledge.

Millsap, R. E., & Yun-Tein, J. (2004). Assessing factorial invariance in ordered-categorical measures. *Multivariate Behavioral Research, 39*(3), 479–515. doi:10.1207/s15327906mbr3903_4

Muthén, B. (1984). A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators. *Psychometrika, 49*(1), 115–132.

Satorra, A., & Bentler, P. M. (2001). A scaled difference chi-square test statistic for moment structure analysis. *Psychometrika, 66*(4), 507–514. doi:10.1007/BF02296192

Wu, H., & Estabrook, R. (2016). Identification of confirmatory factor analysis models of different levels of invariance for ordered categorical outcomes. *Psychometrika, 81*(4), 1014–1045. doi:10.1007/s1133601695060

See Also

`cfa_1` and `cfa_k` for the single-group measurement models; `reliability_omega` for the reliability of the composite the model justifies; `compare_cov_structures` for covariance-structure comparisons outside the factor model; `lavTestScore` for the score test that localizes a failed rung.

Other multivariate and latent variable methods: `average_variance_extracted()`, `bifactor_indices()`, `cfa_1()`, `cfa_2()`, `cfa_k()`, `ci_eigenvalue()`, `common_method_marker()`, `common_method_single_factor()`, `dmacs()`, `ecvi()`, `htmt()`, `irt_grm()`, `irt_information()`, `measurement_alignment()`, `procrustes_phi()`, `simple_structure()`

Examples

```
# Do the two schools measure the verbal and the reasoning construct
# in the same way? (Holzinger & Swineford, bundled.) The measurement
# model is a named list of factors; for a single factor, name its
# indicators with items = instead, or give lavaan syntax, as the last
# example does.
data(holzinger_swineford)
hs_factors <- list(
  verbal    = c("t6_paragraph_comprehension", "t7_sentence",
                "t9_word_meaning"),
  deduction = c("t20_deduction", "t22_problem_reasoning",
                "t23_series_completion"))

# The whole ladder, configural through metric, scalar, and strict.
# Configural invariance asks whether the same pattern of loadings
# holds at both schools; metric adds the constraint that the loadings
# are equal across schools, and it is the rung that has to hold before
# a relation involving one of these factors is compared across them;
# scalar equates the intercepts as well, which comparing factor means
# requires; strict equates the residual variances. Each row from the
# second on tests its rung against the one below it.
mi <- measurement_invariance(holzinger_swineford, hs_factors,
                             group = "school")
mi

# The fitted models travel with the table, so localizing a failed rung
# costs no refitting.
names(attr(mi, "fits"))
```

```
# The broom verbs: one row per rung of the ladder, and the model-level
# summary (estimator, test flavor, fit index flavor).
generics::tidy(mi)
generics::glance(mi)

# Partial invariance frees one loading across the schools at every
# rung (Byrne, Shavelson, & Muthén, 1989), so the metric rung costs
# one degree of freedom fewer than it did above. Naming a leading
# subset of the rungs in levels = stops the ladder there, which keeps
# this and the next two examples quick; a reported analysis climbs
# the whole ladder.
measurement_invariance(holzinger_swineford, hs_factors,
                       group = "school",
                       group_partial = "verbal =~ t7_sentence",
                       levels = c("configural", "metric"))

# With the indicators coded as ordered categories, ordered = TRUE puts
# the thresholds rung first, because thresholds rather than intercepts
# carry the location information there, and makes the rung-to-rung
# tests the scaled difference tests; the "test" attribute names the
# test that was used. The ladder stops at the thresholds rung here;
# metric, scalar, and strict follow it.
hs_ordered <- holzinger_swineford
for (item in unlist(hs_factors, use.names = FALSE)) {
  hs_ordered[[item]] <- as.integer(cut(
    holzinger_swineford[[item]],
    breaks = quantile(holzinger_swineford[[item]],
                      c(0, .25, .5, .75, 1)),
    include.lowest = TRUE))
}
mi_ordered <- measurement_invariance(hs_ordered, hs_factors,
                                     group = "school", ordered = TRUE,
                                     levels = c("configural",
                                                "thresholds"))
mi_ordered
attr(mi_ordered, "test")

# The measurement model can also be given as lavaan syntax, and
# missing = "fiml" fits the ladder by full information maximum
# likelihood when continuous items have incomplete cases; here twenty
# scores on one item are set to missing first. With three indicators
# the single factor is just identified in each group, so the
# configural row is saturated (zero degrees of freedom, exact fit)
# and the metric row carries the test.
hs_missing <- holzinger_swineford
set.seed(113)
hs_missing$t7_sentence[sample(nrow(hs_missing), 20)] <- NA
measurement_invariance(
  hs_missing,
  model = "verbal =~ t6_paragraph_comprehension + t7_sentence +
           t9_word_meaning",
  group = "school", missing = "fiml",
  levels = c("configural", "metric"))
```

`mediate` *Mediation Analysis With Bootstrap Confidence Intervals*

Description

Estimates the simple mediation model, a predictor X affecting an outcome Y directly and through a mediator M, and reports the indirect, direct, and total effects with confidence intervals in one tidy table. The indirect effect ab is the product of the $X \to M$ path and the $M \to Y$ path (holding X), and its sampling distribution is skewed, which is why the default interval is the percentile bootstrap rather than a normal approximation; the bias-corrected and accelerated (BCa) bootstrap, the Monte Carlo (parametric simulation) interval, and the Sobel normal-theory interval are available for comparison. This is the analysis counterpart of the planning functions `ss_aipe_indirect_effect` and `ss_power_indirect_effect`.

Usage

```
mediate(
  data,
  x,
  m,
  y,
  covariates = NULL,
  ci_method = c("boot_percentile", "boot_bca", "monte_carlo", "sobel"),
  B = 2000,
  conf_level = 0.95,
  seed = NULL
)
```

Arguments

`data`	A `data.frame` containing the variables.
`x`, `m`, `y`	Names (single character strings) of the predictor, the mediator, and the outcome columns in `data`.
`covariates`	Optional character vector of covariate column names, entered in both the mediator and the outcome models.
`ci_method`	Confidence interval method for the indirect effect: `"boot_percentile"` (default), `"boot_bca"`, `"monte_carlo"` (simulate a and b from their joint normal approximation; MacKinnon, Lockwood, & Williams, 2004), or `"sobel"` (the first-order normal-theory interval; reported for comparison, not recommended for inference). Direct and total effects always carry their ordinary t-based intervals.
`B`	Number of bootstrap or Monte Carlo replications. Defaults to 2000; published analyses often use 5000 or more, and the BCa interval in particular rewards a large `B` (see *Details*).

conf_level	Confidence level. Defaults to 0.95.
seed	Optional integer seed for the resampling, used locally (the caller's random number generator state is restored on exit). Default NULL leaves the random number generator state alone.

Details

The model is the standard pair of regressions

$$M = i_M + aX + \mathbf{g}'\mathbf{C} + e_M, \qquad Y = i_Y + c'X + bM + \mathbf{h}'\mathbf{C} + e_Y,$$

with $\mathbf{C}$ the optional covariates. The indirect effect is ab, the direct effect c', and the total effect $c = c' + ab$ (an identity in linear models with the same cases, which the implementation exploits as an internal consistency check).

Bootstrap intervals resample cases (rows) with replacement B times, refitting both regressions in each resample (Efron & Tibshirani, 1993). "boot_percentile" takes the interval limits from the empirical quantiles of the bootstrapped ab estimates; it is not forced to be symmetric about the estimate, which is the point for a skewed sampling distribution. "boot_bca" (the bias-corrected and accelerated interval) additionally adjusts the two quantile positions for median bias, estimated from the bootstrap distribution, and for the rate at which the variance of the estimator changes with the parameter, the acceleration, estimated by the jackknife (which adds *N* extra pairs of fits); the adjustments make it second-order accurate where the percentile interval is first-order accurate (DiCiccio & Efron, 1996). Because the adjusted quantile positions sit farther into the tails of the bootstrap distribution, the BCa interval benefits more than the percentile interval does from a B well above the default. Each resample refits the two regressions by least squares, a closed-form fit with no iterative estimation, so in ordinary data all B replications enter the interval. A degenerate resample (one whose refit is rank deficient, possible with a near-constant predictor) returns no indirect effect; such replications are dropped with a warning stating how many, and the interval is computed from the replications that returned a value. The Sobel standard error is computed by var_indirect_effect. The proportion mediated is one of the effect size measures for mediation models surveyed by Preacher and Kelley (2011); its instability when the total effect is small is why it is reported as NA near a zero total effect. Listwise deletion is applied to the analysis variables; for full information maximum likelihood under missingness, fit the model in **lavaan** (see mlmr for the package's FIML front end philosophy).

Mediation language implies causal structure: with observational data the estimates are conditional associations, and the causal reading requires the usual no-unmeasured-confounding assumptions for both the $X \to M$ and $M \to Y$ links (MacKinnon, 2008). The function computes; the design earns the interpretation.

Value

A data.frame (class dmar_tbl) with rows indirect_effect (with its ci_method interval), direct_effect, total_effect, the paths a and b, their standard errors (se_indirect per Sobel, se_a, se_b), the interval limits (indirect_lower / indirect_upper, direct_lower / direct_upper, total_lower / total_upper), proportion_mediated (ab/c; NA when the total effect is near zero, where the ratio is unstable), N, and B. The interval method is recorded in the "ci_method" attribute and the confidence level in "conf_level".

Author(s)

Ken Kelley <kkelley@nd.edu>

References

DiCiccio, T. J., & Efron, B. (1996). Bootstrap confidence intervals. *Statistical Science, 11*(3), 189–228.

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, NY: Chapman & Hall/CRC.

MacKinnon, D. P. (2008). *Introduction to statistical mediation analysis*. Erlbaum.

MacKinnon, D. P., Lockwood, C. M., & Williams, J. (2004). Confidence limits for the indirect effect: Distribution of the product and resampling methods. *Multivariate Behavioral Research, 39*(1), 99–128. doi:10.1207/s15327906mbr3901_4

Preacher, K. J., & Hayes, A. F. (2008). Asymptotic and resampling strategies for assessing and comparing indirect effects in multiple mediator models. *Behavior Research Methods, 40*(3), 879–891. doi:10.3758/BRM.40.3.879

Preacher, K. J., & Kelley, K. (2011). Effect size measures for mediation models: Quantitative strategies for communicating indirect effects. *Psychological Methods, 16*(2), 93–115. doi:10.1037/a0022658

See Also

`ss_aipe_indirect_effect` and `ss_power_indirect_effect` for planning the study this function analyzes; `var_indirect_effect` for the Sobel variance.

Other mediation: `mediation_mbco()`, `ss_power_indirect_effect()`

Examples

```
# Simulated mediation: X raises M (a = .5), M raises Y (b = .4), and a
# little direct effect remains (c' = .2).
set.seed(113)
n <- 200
x <- rnorm(n)
m <- 0.5 * x + rnorm(n, 0, sqrt(1 - 0.25))
y <- 0.2 * x + 0.4 * m + rnorm(n, 0, 0.8)
d <- data.frame(x = x, m = m, y = y)

# The percentile bootstrap is the default and is what a reported
# analysis would use. Each resample refits both regressions, so
# B = 200 keeps the example quick; a reported interval deserves the
# default B = 2000 or more.
mediate(d, x = "x", m = "m", y = "y", B = 200, seed = 113)

# The Monte Carlo interval draws a and b from their joint normal
# approximation instead of refitting, so it stays quick even at a
# large B. Its limits sit close to the bootstrap limits here.
mediate(d, x = "x", m = "m", y = "y", ci_method = "monte_carlo",
        B = 10000, seed = 113)
```

```
# The Sobel interval is closed form. It assumes the product ab is
# normally distributed, which is why it is reported for comparison
# rather than used for inference; its B row is NA because no
# replications are drawn.
mediate(d, x = "x", m = "m", y = "y", ci_method = "sobel")
```

`mediation_mbco` *Mediation Analysis via Model-Based Constrained Optimization*

Description

Tests hypotheses about mediation effects with the model-based constrained optimization (MBCO) procedure of Tofighi and Kelley (2020): a likelihood ratio test that compares the full mediation model to a null model fit subject to the nonlinear constraint that the effect of interest (an indirect effect, a total effect, or any smooth function of path coefficients) equals zero. The model is specified in **lavaan** syntax but is fit with **OpenMx**, whose optimizers support the nonlinear equality constraints the procedure requires (see Details). The function enumerates the mediation effects implied by the model (the total effect, the direct effect, the total indirect effect, and every specific indirect pathway from `x` to `y`), reports each with a confidence interval, and tests each with the MBCO likelihood ratio statistic and its *p*-value. Models may include observed or latent variables, one or many mediators, in parallel or in sequence, and raw data or summary statistics may be supplied. With a grouping variable the model becomes a multiple-group SEM: every effect is estimated within each group and the between-group difference of each effect is tested, which is moderated mediation with a categorical moderator. With a numeric `moderator`, interactions stated in the model are probed: conditional effects at chosen moderator values, the index of moderated mediation, and a joint constancy test, all with MBCO likelihood ratio inference (see Details).

Usage

```
mediation_mbco(
  model,
  data = NULL,
  S = NULL,
  M = NULL,
  N = NULL,
  group = NULL,
  x = NULL,
  y = NULL,
  moderator = NULL,
  probe_values = NULL,
  hypotheses = NULL,
  ci_method = c("profile_likelihood", "monte_carlo", "wald"),
  conf_level = 0.95,
  B = 10000,
  optimizer = c("SLSQP", "CSOLNP", "NPSOL"),
  seed = NULL
)
```

Arguments

model	A character string with the model in **lavaan** syntax (regressions ~, latent variable definitions =~, residual covariances ~~, optionally labeled coefficients and defined quantities via :=), or a fitted **lavaan** object whose parameter table and data are then reused. Parameter labels must be valid R names (e.g., b1, not 1b).
data	A data.frame of raw data containing the observed variables. Rows with missing values are retained and handled by full information maximum likelihood. Supply either data or S (with N), not both.
S	A covariance matrix of the observed variables (with dimnames naming the variables), used with N and optionally M when raw data are not available. Complete-data maximum likelihood depends on the data only through the sample means and covariance matrix, so an analysis from summary statistics reproduces the raw-data analysis exactly (see Details). For a multiple-group analysis, a named list of such matrices, one per group.
M	Optional named numeric vector of variable means accompanying S (for multiple groups, a named list of such vectors). When omitted, means of zero are used; intercepts are then uninteresting but every effect, test, and confidence interval is unaffected.
N	Total sample size. Required with S; ignored for raw data. For a multiple-group analysis, a vector with one sample size per group in S (matched by name when named).
group	Optional name (single character string) of the grouping variable in data for a multiple-group analysis. With summary input, supply S, M, and N as lists instead and leave group alone (it then merely names the constructed grouping column). See the Details section on multiple groups, including how lavaan's labeling rules decide which parameters differ by group.
x, y	Names (single character strings) of the focal predictor and the outcome between which effects are traced. When both are omitted the function looks for a unique variable with no incoming regression path (the source) and a unique variable with no outgoing regression path (the outcome); if either is ambiguous an error asks for x and y explicitly. When the model defines quantities via := or hypotheses are supplied, x and y may be omitted entirely and only those quantities are estimated and tested.
moderator	Optional name (single character string) of a numeric moderator variable, turning on probing: every pathway effect that involves an interaction with the moderator is estimated at each probe value, the change in the effect per unit moderator is reported (for a pathway moderated in one place, the index of moderated mediation), and a constancy test asks whether the pathway effect depends on the moderator at all. State the interaction in the model syntax with a : term (e.g., m ~ x + w + x:w) or include a precomputed product column among the predictors; both are recognized. A categorical moderator goes through group instead.
probe_values	Numeric vector of moderator values at which moderated effects are estimated; names, when given, label the rows. Defaults to the moderator's mean and one standard deviation either side (labeled low, mean, high), or to the two observed values of a two-valued moderator.

`hypotheses` Optional named character vector of additional quantities to estimate and test against zero, written as R expressions in the parameter labels and effect names (e.g., `c(difference = "indirect_via_imagery - indirect_via_repetition")` tests the equality of two specific indirect effects). A named *list* may mix such scalar expressions with character vectors of several expressions; a multi-expression element is tested jointly (all constrained to zero at once, with as many degrees of freedom as expressions), and its row reports the test alone, with no scalar estimate.

`ci_method` Confidence interval method for every reported effect: `"profile_likelihood"` (default; Neale & Miller, 1997, computed by **OpenMx**), `"monte_carlo"` (simulate the coefficients from their joint normal approximation and form percentile intervals of the effect; MacKinnon, Lockwood, & Williams, 2004), or `"wald"` (delta method symmetric interval; reported for comparison, not recommended for indirect effects, whose sampling distributions are skewed).

`conf_level` Confidence level. Defaults to 0.95.

`B` Number of Monte Carlo replications when `ci_method = "monte_carlo"`. Defaults to 10000.

`optimizer` Optimizer used by **OpenMx** for all fits: `"SLSQP"` (default), `"CSOLNP"`, or `"NPSOL"`. The CRAN build of **OpenMx** ships SLSQP and CSOLNP; NPSOL, the optimizer used in Tofighi and Kelley (2020), is available only in the build distributed from the OpenMx website.

`seed` Optional integer seed for the Monte Carlo interval, used locally (the caller's random number generator state is restored on exit). Default `NULL` leaves the random number generator state alone.

Details

The MBCO procedure. A hypothesis about mediation is recast as a model comparison (Maxwell, Delaney, & Kelley, 2027). The full model estimates all path coefficients freely. The null model is the same model estimated subject to the constraint that the effect of interest is zero, for example H_0 : $\beta_1\beta_2 = 0$ for the indirect effect of X on Y through one mediator. Writing $D = -2\log\text{likelihood}$ for the deviance of each fitted model, the test statistic is

$$\text{LRT}_{\text{MBCO}} = D_{\text{null}} - D_{\text{full}},$$

which has a large sample chi square distribution with degrees of freedom equal to the number of constraints (here 1) when the null hypothesis is true (Wilks, 1938). The resulting *p*-value is a continuous measure of the compatibility of the data with the null model, not merely a reject or fail-to-reject decision, and in the simulations of Tofighi and Kelley (2020) the test controls the Type I error rate more robustly than the tests based on confidence intervals in common use, especially when the indirect effect is truly zero.

Why the constraint is nonlinear. Null hypotheses about single parameters are linear constraints: fixing $\beta_3 = 0$ restricts the parameter space to a flat hyperplane, something every structural equation modeling program does by fixing the parameter and refitting. The null hypothesis of no mediation is different in kind. The constraint $\beta_1\beta_2 = 0$ involves a product of parameters, so it is a nonlinear function of the parameter vector, and its solution set is not a hyperplane but the union of two hyperplanes: the set where $\beta_1 = 0$ (with β_2 free) and the set where $\beta_2 = 0$ (with β_1 free). Mediation

is absent in infinitely many ways, and no single fixed parameter expresses them all: fixing $\beta_1 = 0$ alone tests a more restrictive hypothesis than $H_0 : \beta_1\beta_2 = 0$. Maximizing the likelihood over such a constraint set requires an optimizer that handles general nonlinear equality constraints (sequential quadratic programming methods such as SLSQP and NPSOL, or CSOLNP); among R packages for structural equation modeling, **OpenMx** provides them, which is why the model is fit there even though it is specified in **lavaan** syntax. The same geometry explains why Wald-type (delta method) tests of a product behave badly near the null: at $\beta_1 = \beta_2 = 0$ the two hyperplanes intersect, the gradient of $\beta_1\beta_2$ vanishes, and the usual normal approximation for the product breaks down. The MBCO procedure sidesteps that approximation by comparing maximized likelihoods directly.

Local solutions and the search strategy. Because the null set is a union of surfaces, the constrained deviance surface generally has one local minimum per branch (one where the mediator does not respond to X, one where the outcome does not respond to the mediator, and so on along longer chains). A constrained optimizer started from the full-model estimates can converge to whichever branch is nearest rather than to the branch that fits best. The likelihood ratio test is defined by the globally best-fitting null model, so `mediation_mbco()` refits each null model from several starting configurations (the full-model estimates, and the estimates with each coefficient entering the constrained effect set to zero in turn), verifies that each candidate solution actually satisfies the constraint, and keeps the feasible solution with the smallest deviance. In the memory example below this matters: for the single mediator model the best-fitting null model sets the imagery-to-recall path to zero ($\mathrm{LRT}_{\mathrm{MBCO}} = 71.31$, the statistic the example reports), while the branch that sets the instruction-to-imagery path to zero fits worse ($\mathrm{LRT}_{\mathrm{MBCO}} = 179.02$). Tofighi and Kelley (2020) report 175.77 for this example, a value from that worse-fitting branch, on which their optimizer stopped; their statistic also differs from the 179.02 here because the example runs from the published (rounded) summary statistics of their Table 1 rather than the full-precision moments (which give 72.54 and 175.77 for the two branches). Either way the null model is overwhelmingly incompatible with the data, so the substantive conclusion is the same.

Effects estimated and tested. With `x` and `y` resolved, the function enumerates every directed pathway from `x` to `y` along regression (`~`) paths. Each pathway through at least one intermediate variable contributes a specific indirect effect, the product of its path coefficients, reported as `indirect_via_` followed by the intermediate variable names. The direct effect is the `x` to `y` coefficient when that path is in the model. The total indirect effect is the sum of the specific indirect effects (reported when there are two or more), and the total effect is the direct effect plus the total indirect effect (reported when the direct path is in the model). Quantities defined in the model syntax via `:=` and any `hypotheses` are estimated and tested the same way, so contrasts of indirect effects, proportions mediated, or any other smooth function of parameters can be examined; each is tested against zero, so write an equality of two effects as their difference. Every reported row carries its estimate, a delta method standard error (via `mxSE`), the `ci_method` confidence interval, and the MBCO likelihood ratio test with its degrees of freedom and *p*-value.

The parallel two-mediator model. Tofighi and Kelley (2020) continue the memory example with imagery and repetition as parallel mediators and a residual covariance between them, asking whether the indirect effect through repetition is zero and whether the two specific indirect effects differ (their Research Questions 2 and 3). That analysis is the single-mediator call of the examples with the parallel model in place of the single one and the contrast supplied through `hypotheses`:

```
parallel <- "
  imagery    ~ b1*instruction
  repetition ~ b3*instruction
  recall     ~ b2*imagery + b4*repetition + b5*instruction
```

```
  imagery ~~ repetition
"
mediation_mbco(parallel, S = S_tk, M = M_tk, N = 369,
               x = "instruction", y = "recall",
               hypotheses = c(imagery_minus_repetition =
                 "indirect_via_imagery - indirect_via_repetition"),
               ci_method = "monte_carlo", seed = 113)
```

Six effects are reported, so six constrained null models are fit, about twice the cost of the single-mediator analysis. The indirect effect through repetition is near zero (the paper reports $\mathrm{LRT}_{\mathrm{MBCO}} = 0.083$, $p = .773$), while the contrast shows the imagery pathway is larger (the paper reports $\mathrm{LRT}_{\mathrm{MBCO}} = 25.828$, difference = 2.222, SE = 0.445).

Choosing the confidence interval. The default profile likelihood interval inverts the likelihood ratio test for the effect itself (Neale & Miller, 1997), so its limits are free to sit asymmetrically about the estimate, as the skewed sampling distribution of a product of coefficients calls for, and the interval and the MBCO test tell the same story. Each bound is a constrained search of its own, which is what makes it the expensive choice. The Monte Carlo interval is the inexpensive alternative that also accommodates the skewness: `B` coefficient vectors are drawn from the joint normal approximation of the estimates, the effect is evaluated in each draw, and the limits are the empirical $(\alpha/2, 1-\alpha/2)$ quantiles of those `B` values (MacKinnon, Lockwood, & Williams, 2004). Ask for it when the profile searches are slow, when a profile bound fails to converge, or when comparing with a published analysis that reports one, as Tofighi and Kelley (2020) do for the memory data in the examples. The Wald interval is the symmetric delta method interval and is reported only for comparison.

Multiple groups (moderated mediation across groups). With `group` (or list-form `S`, `M`, `N`), the model is fit as a multiple-group SEM: the same structure in every group, with group-specific parameters. Every enumerated effect is then estimated within each group (its term carries the group label as a suffix), and for each effect the difference from the reference group (the first group label) is estimated and tested with the same constrained-optimization machinery, since a difference of two products is itself a nonlinear function of the parameters. That difference test is moderated mediation with a categorical moderator: "does the indirect effect differ across groups?" is exactly the between-group contrast of the conditional indirect effects. lavaan's labeling rules decide what varies: a single label such as `b1` on a path applies to every group and therefore imposes cross-group equality (the corresponding difference is identically zero and its row is dropped); to let a path differ by group, leave it unlabeled or give per-group labels with the vector form `c("b1_f", "b1_m")*x`. With more than two groups, each non-reference group is compared with the reference group. The call below fits the simple model in both groups of a data frame `d` whose `condition` column takes two values, and reports every effect per group beside its between-group difference; leaving the paths unlabeled lets them differ by group.

```
mediation_mbco("m ~ x
                y ~ m + x",
               data = d, group = "condition", x = "x", y = "y",
               ci_method = "wald")
```

Each reported row costs its own constrained null model fit, so this two-group analysis fits nine null models where the single-group analysis fits three.

Moderated mediation, probed. With `moderator`, every regression coefficient along a pathway becomes a linear function of the moderator wherever the model contains the matching interaction

(stated as `x:w` in the syntax, or as a product column among the predictors). A pathway effect, the product of its edge coefficients, is then a polynomial in the moderator, and the function derives that polynomial symbolically. Three kinds of rows follow for each moderated effect. First, the conditional effect at each probe value, each with its own confidence interval and MBCO test. Second, the polynomial's moderator coefficients: for a pathway moderated in one place the single such coefficient is exactly the index of moderated mediation (Hayes, 2015), here tested by likelihood ratio rather than bootstrap; a pathway moderated in several places gets one row per power of the moderator. Third, for a pathway moderated in several places (so the conditional effect is curved in the moderator), a joint constancy test of all moderator coefficients at once, with as many degrees of freedom as constraints, asking whether the pathway effect depends on the moderator at all; as a joint test its row reports no scalar estimate. Unmoderated effects in the same model keep their single rows. The null set of a conditional-effect constraint is again a union of branches (one edge's conditional coefficient or another's must vanish at the probed value), and the null fits are started on each branch, and at the model with all interactions removed for the moderation tests, so the reported statistics reflect the best-fitting null models. For example:

```
mediation_mbco("m ~ x + w + x:w
                y ~ m + x + w",
               data = d, x = "x", y = "y", moderator = "w")
```

`plot_mediation_mbco` draws the same conditional effects as curves over the moderator's whole range with a confidence band, the visual companion to the probe rows. Bespoke conditional quantities can still be written directly with `:=` definitions or `hypotheses` when the built-in probing does not cover them.

Moderated mediation and mediated moderation. Moderated mediation asks whether an indirect effect depends on a moderator W: the conditional indirect effect varies with w (Muller, Judd, & Yzerbyt, 2005; Preacher, Rucker, & Hayes, 2007). Mediated moderation asks whether an observed $X \times W$ interaction on Y is transmitted through the mediator. The two share their algebra: with first-stage moderation $M = a_1X + a_2W + a_3XW + e_M$ and $Y = bM + \ldots$, the quantity a_3b is at once the index of moderated mediation (Hayes, 2015) and the indirect effect of the product term through M; what differs is the question and the reporting emphasis. This function reports the conditional-indirect-effect framing: the probe rows and the moderation rows above. For a categorical moderator, use `group`.

Refusals, warnings, and judgment calls. The function stops where a computed number would be mislabeled: a nonrecursive (feedback) regression structure, categorical-endogenous syntax (thresholds), explicit `==` constraints, a model with no indirect pathway between `x` and `y`. It warns where the data make trouble detectable: an endogenous variable with only two distinct values (a binary mediator or outcome, where the product of coefficients is not the causal indirect effect), an interaction term among the predictors that no declared `moderator` accounts for (declaring the moderator resolves the warning by probing the moderation), an interaction included without its matching main effect (the principle of marginality), null models that converge imperfectly, profile bounds that fail. What it cannot check is left to the analyst, and stated rather than assumed: the no omitted confounder assumption, linearity, and the causal direction of the arrows come from the design, not from the fit.

Information criteria. The columns `delta_aic` and `delta_bic` report AIC and BIC for the null model minus the same criterion for the full model; positive values favor the full model (the effect improves fit by more than the parsimony penalty). A scalar equality constraint reduces the effective number of free parameters by one, so the differences equal $\mathrm{LRT}_{\mathrm{MBCO}} - 2$ for the AIC and

$\mathrm{LRT}_{\mathrm{MBCO}} - \log N$ for the BIC. (The **OpenMx** summary counts the same number of estimated parameters in both models, so its printed AIC difference equals the likelihood ratio statistic; DMAR counts the constraint against the null model, consistent with the degrees of freedom of the test.)

Change in explained variance. Following the reporting recommendation of Tofighi and Kelley (2020), the function computes R^2 for every endogenous variable under the full model (the `"R2"` attribute) and the drop in each R^2 under every null model (the `"delta_R2"` attribute, variables by tested effects), so the fit cost of removing an effect can be read as a change in effect size and not only as a test statistic.

Summary statistics input. With complete data the multivariate normal log likelihood depends on the data only through the sample means and the sample covariance matrix. When `S` (and optionally `M`) is supplied, the function therefore constructs an internal data set with exactly those moments (via `mvrnorm` with `empirical = TRUE`) and proceeds as with raw data; the estimates, likelihood ratio tests, and confidence intervals are identical to what the raw data would give, whatever internal data set realizes the moments. `S` is treated as the unbiased (divisor $N - 1$) covariance matrix, the form reported in articles. This is how a published mediation analysis can be reproduced, and its hypotheses re-tested with the MBCO procedure, from a table of descriptive statistics alone.

Assumptions. The causal reading of any mediation analysis rests on the no omitted confounder assumption for the predictor to mediator and mediator to outcome relations, in addition to the usual distributional assumptions (residuals multivariate normal, linear relations, no treatment by mediator interaction); randomizing X supports the first link but not the second (MacKinnon, 2008; Tofighi & Kelley, 2020). With a binary randomized X the variable enters the model as numeric 0/1, its exogenous variance freely estimated; the normality assumption applies to the residuals of the endogenous variables.

Value

A `data.frame` (classes `dmar_mediation_mbco`, `dmar_tbl`) with one row per effect and columns `pathway` (the traced pathway or defining expression), `term`, `estimate`, `se` (delta method), `ci_lower`, `ci_upper`, `lrt` (the MBCO likelihood ratio statistic), `df`, `p_value`, `delta_aic`, and `delta_bic`. A joint test row (a several-constraint moderation or hypothesis test) reports the test columns with `df` equal to the number of constraints; its `estimate`, `se`, and interval are `NA` because no single number summarizes several constraints. Attributes: `"conf_level"`, `"ci_method"`, `"optimizer"`, `"x"`, `"y"`, `"N"`, `"deviance"`, `"aic"`, `"bic"`, `"n_par"` (full model fit information), `"groups"` (the group labels, reference first; multiple-group fits only), `"R2"` (named vector, endogenous variables under the full model, per group when grouped), `"delta_R2"` (matrix of full minus null R^2, variables by tested effects), `"moderation"` (for a moderated analysis: the moderator name, the probe values, and each moderated effect's moderator polynomial, which is what `plot_mediation_mbco` draws), and `"mx_model"` (the fitted **OpenMx** full model, an escape hatch for further OpenMx work). Use `tidy()` and `glance()` for broom-style views.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Tofighi, D., & Kelley, K. (2020). Improved inference in mediation analysis: Introducing the model-based constrained optimization procedure. *Psychological Methods, 25*(4), 496–515. doi:10.1037/

met0000259

Tofighi, D., & Kelley, K. (2020). Indirect effects in sequential mediation models: Evaluating methods for hypothesis testing and confidence interval formation. *Multivariate Behavioral Research, 55*(2), 188–210. doi:10.1080/00273171.2019.1618545

Hayes, A. F. (2015). An index and test of linear moderated mediation. *Multivariate Behavioral Research, 50*(1), 1–22. doi:10.1080/00273171.2014.962683

MacKinnon, D. P. (2008). *Introduction to statistical mediation analysis*. Erlbaum.

MacKinnon, D. P., Lockwood, C. M., & Williams, J. (2004). Confidence limits for the indirect effect: Distribution of the product and resampling methods. *Multivariate Behavioral Research, 39*(1), 99–128. doi:10.1207/s15327906mbr3901_4

MacKinnon, D. P., Valente, M. J., & Wurpts, I. C. (2018). Benchmark validation of statistical models: Application to mediation analysis of imagery and memory. *Psychological Methods, 23*(4), 654–671. doi:10.1037/met0000174

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Muller, D., Judd, C. M., & Yzerbyt, V. Y. (2005). When moderation is mediated and mediation is moderated. *Journal of Personality and Social Psychology, 89*(6), 852–863. doi:10.1037/0022-3514.89.6.852

Neale, M. C., Hunter, M. D., Pritikin, J. N., Zahery, M., Brick, T. R., Kirkpatrick, R. M., Estabrook, R., Bates, T. C., Maes, H. H., & Boker, S. M. (2016). OpenMx 2.0: Extended structural equation and statistical modeling. *Psychometrika, 81*(2), 535–549. doi:10.1007/s1133601494358

Neale, M. C., & Miller, M. B. (1997). The use of likelihood-based confidence intervals in genetic models. *Behavior Genetics, 27*(2), 113–120. doi:10.1023/A:1025681223921

Preacher, K. J., Rucker, D. D., & Hayes, A. F. (2007). Addressing moderated mediation hypotheses: Theory, methods, and prescriptions. *Multivariate Behavioral Research, 42*(1), 185–227. doi:10.1080/00273170701341316

Wilks, S. S. (1938). The large-sample distribution of the likelihood ratio for testing composite hypotheses. *Annals of Mathematical Statistics, 9*(1), 60–62. doi:10.1214/aoms/1177732360

See Also

`plot_mediation_mbco` for the conditional-effect display of a moderated analysis; `mediate` for the regression-based simple mediation model with bootstrap intervals; `ss_aipe_indirect_effect` and `ss_power_indirect_effect` for planning; `var_indirect_effect` for the delta method variance.

Other mediation: `mediate()`, `ss_power_indirect_effect()`

Examples

```
# Replicate the memory experiment analyses of Tofighi and Kelley
# (2020) from the summary statistics in their Table 1 (data from
# MacKinnon, Valente, & Wurpts, 2018; N = 369). Instruction is 1 for
# imagery rehearsal instructions and 0 for repetition instructions.
# Requires the OpenMx and lavaan packages to be installed.
vars <- c("instruction", "imagery", "repetition", "recall")
sds  <- c(0.50, 2.96, 2.84, 3.40)
```

```
R_tk <- matrix(c(1.00,  .62, -.67,  .32,
                  .62, 1.00, -.56,  .51,
                 -.67, -.56, 1.00, -.28,
                  .32,  .51, -.28, 1.00), 4, 4,
               dimnames = list(vars, vars))
S_tk <- outer(sds, sds) * R_tk
M_tk <- c(instruction = 0.51, imagery = 5.66, repetition = 6.08,
          recall = 12.07)

# Single-mediator model: instruction -> imagery -> recall. Every
# reported effect costs its own constrained null model fit in OpenMx,
# which is where the run time goes; the Monte Carlo interval itself
# is inexpensive at any B. The indirect effect is about 2.1 words
# (the paper reports 2.121, SE = 0.276, 95% Monte Carlo CI
# [1.600, 2.682]), and its likelihood ratio statistic of 71.31 is the
# value discussed under Details.
single <- "
  imagery ~ b1*instruction
  recall  ~ b2*imagery + b3*instruction
"
mediation_mbco(single, S = S_tk, M = M_tk, N = 369,
               x = "instruction", y = "recall",
               ci_method = "monte_carlo", seed = 113)

# The parallel two-mediator model of the same paper, with the contrast
# of its two specific indirect effects, is described under Details;
# it is fit the same way from the same summary statistics, adding the
# 'hypotheses' argument. Raw data go in through 'data' rather than
# 'S', 'M', and 'N'; adding 'group' fits the model in every group and
# tests the between-group difference of each effect, which is
# moderated mediation with a categorical moderator (see Details). A
# continuous moderator goes in through 'moderator', and the help page
# for plot_mediation_mbco fits that analysis from a data frame and
# draws the conditional effects it estimates.
```

`meta_contrast` *Contrast Among Study Effect Sizes (Rosenthal-Rubin)*

Description

Tests a focused hypothesis about *differences* among independent study effect sizes by the method of Rosenthal and Rubin (1982): given effects y_i with sampling variances v_i and contrast weights λ_i summing to zero,

$$z = \frac{\sum \lambda_i y_i}{\sqrt{\sum \lambda_i^2 v_i}}$$

is referred to the standard normal. This is how a meta-analyst asks a pointed moderator question ("do the effects decline with weeks of prior teacher-student contact?") rather than the diffuse heterogeneity question ("do the effects differ at all?"). Raudenbush (1984) used exactly this test for the teacher expectancy literature, with weights inversely proportional to weeks of prior contact.

Usage

```
meta_contrast(yi, vi, weights, center = TRUE)
```

Arguments

yi	Numeric vector of study effect sizes (any metric whose sampling distribution is approximately normal; standardized mean differences and Fisher's Z correlations qualify).
vi	Sampling variances of yi, one per study.
weights	Contrast weights, one per study. If they do not already sum to zero they are mean-centered (with a message) when center = TRUE, the convenient route for weights built from a moderator such as 1 / (weeks + 2).
center	Logical: mean-center weights that do not sum to zero? Default TRUE.

Details

The two-sided p-value is reported; halve it for a directional hypothesis stated in advance (Raudenbush's $z = 2.75$ carried the one-tailed $p = .003$). Dividing the squared contrast z^2 by the total heterogeneity statistic Q from meta_es gives the proportion of between-study heterogeneity the contrast accounts for, the meta-analytic analog of a contrast's share of the between-group sum of squares.

Value

A data.frame (class dmar_tbl) with the contrast estimate ($\sum \lambda_i y_i$), its se, the z statistic, the two-sided p_value, and k.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Raudenbush, S. W. (1984). Magnitude of teacher expectancy effects on pupil IQ as a function of the credibility of expectancy induction: A synthesis of findings from 18 experiments. *Journal of Educational Psychology, 76*(1), 85–97.

Rosenthal, R., & Rubin, D. B. (1982). Comparing effect sizes of independent studies. *Psychological Bulletin, 92*(2), 500–504.

See Also

meta_es for the pooled effect and the total heterogeneity the contrast partitions; combine_p for combined significance tests; contrast_test for the single-study ANOVA analog.

Other meta-analysis: combine_p(), meta_es(), meta_r(), meta_smd(), plot_forest()

Examples

```
# Raudenbush (1984): do expectancy effects decline with weeks of prior
# teacher-student contact? Weights inversely proportional to weeks + 2,
# study-level data (Pellegrini & Hicks merged), d variances from the
# standard large-sample formula.
data(teacher_expectancy)
study <- teacher_expectancy[-c(4, 5), ]
d  <- append(study$d, 0.52, after = 3)
wk <- append(study$weeks, 0, after = 3)
ne <- append(study$n_experimental, 22, after = 3)
nc <- append(study$n_control, 22, after = 3)
v  <- (ne + nc) / (ne * nc) + d^2 / (2 * (ne + nc))
meta_contrast(d, v, weights = 1 / (wk + 2))
# z near 2.75: the better teachers knew their pupils, the smaller the
# expectancy effect (one-tailed p = .003 in the paper).
```

---

meta_es *Random Effects Meta-Analysis of Generic Effect Sizes*

Description

Pools independent effect sizes given their sampling variances: the general engine behind meta_smd and meta_r, exposed for any effect metric whose estimates are approximately normal with known variances. The random effects model is the default and the fit reports the full uncertainty picture in one table: the pooled estimate with its confidence interval, the between-study standard deviation tau, the between-study variance tau-squared with its Q-profile confidence interval, I-squared with an interval mapped from the tau-squared limits, H-squared, Cochran's Q test, and, always, a prediction interval for the effect in a new study. Reporting the prediction interval by default is deliberate: when heterogeneity is real, the confidence interval for the average effect understates what the next study will show, and the package treats "where will the next study land" as part of the answer, not an option.

Usage

```
meta_es(
  yi,
  vi,
  method = c("reml", "pm", "dl", "fe"),
  hartung_knapp = TRUE,
  conf_level = 0.95
)
```

Arguments

yi	Numeric vector of effect sizes, one per independent study.
vi	Sampling variances of yi, one per study.

`method`	Between-study variance estimator: `"reml"` (restricted maximum likelihood, the default), `"pm"` (Paule-Mandel), `"dl"` (DerSimonian-Laird), or `"fe"` (a fixed effect / common effect analysis, which assumes tau-squared is zero and reports no prediction interval).
`hartung_knapp`	Logical: apply the Hartung-Knapp-Sidik-Jonkman small-sample adjustment (the pooled standard error rescaled from the weighted residuals, with a t reference on $k - 1$ degrees of freedom)? Default `TRUE`: with the small numbers of studies typical in psychology and education it keeps the confidence interval near its nominal coverage, where the conventional normal interval is anticonservative. Ignored for `method = "fe"`.
`conf_level`	Confidence level for all intervals. Defaults to 0.95.

Details

The model is $y_i = \mu + u_i + e_i$ with $u_i \sim N(0, \tau^2)$ and $e_i \sim N(0, v_i)$, v_i treated as known. The τ^2 confidence interval inverts the generalized Q statistic (Viechtbauer, 2007); the I-squared interval maps the τ^2 interval through the typical within-study variance of Higgins and Thompson (2002). The prediction interval follows Higgins, Thompson, and Spiegelhalter (2009), using t with $k - 2$ degrees of freedom, and requires at least three studies.

I-squared is reported because readers expect it, but note its well-known limitation: it is a *proportion* of variability, not an amount, so the same tau matched with larger studies yields a larger I-squared. The quantity with direct scientific meaning is tau (the between-study standard deviation, in the metric of `yi`) together with the prediction interval.

Value

A `data.frame` (class `dmar_tbl`) with rows `estimate`, `se`, the test statistic (`t` under Hartung-Knapp, `z` otherwise), `p_value`, `lower_limit` / `upper_limit`, `prediction_lower` / `prediction_upper`, `tau2` with `tau2_lower` / `tau2_upper`, `tau`, `I2` with limits, `H2`, `Q` / `Q_df` / `Q_p`, and `k`. The estimator and adjustment are recorded in the `"method"` and `"hartung_knapp"` attributes.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

DerSimonian, R., & Laird, N. (1986). Meta-analysis in clinical trials. *Controlled Clinical Trials, 7*(3), 177–188.

Hartung, J., & Knapp, G. (2001). On tests of the overall treatment effect in meta-analysis with normally distributed responses. *Statistics in Medicine, 20*(12), 1771–1782. doi:10.1002/sim.791

Higgins, J. P. T., & Thompson, S. G. (2002). Quantifying heterogeneity in a meta-analysis. *Statistics in Medicine, 21*(11), 1539–1558. doi:10.1002/sim.1186

Higgins, J. P. T., Thompson, S. G., & Spiegelhalter, D. J. (2009). A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society: Series A, 172*(1), 137–159. doi:10.1111/j.1467985X.2008.00552.x

Viechtbauer, W. (2007). Confidence intervals for the amount of heterogeneity in meta-analysis. *Statistics in Medicine, 26*(1), 37–52. doi:10.1002/sim.2514

See Also

meta_smd and meta_r for the metric- specific front ends; meta_contrast for focused moderator contrasts; combine_p for combined significance tests; plot_forest to see the studies and the pool together.

Other meta-analysis: combine_p(), meta_contrast(), meta_r(), meta_smd(), plot_forest()

Examples

```
# The teacher expectancy studies (Raudenbush, 1984), pooled in the d
# metric with variances from the standard large-sample formula.
data(teacher_expectancy)
d <- teacher_expectancy$d
n_e <- teacher_expectancy$n_experimental
n_c <- teacher_expectancy$n_control
v <- (n_e + n_c) / (n_e * n_c) + d^2 / (2 * (n_e + n_c))
meta_es(d, v)

# A fixed effect (common effect) analysis of the same studies.
meta_es(d, v, method = "fe")
```

---

meta_r *Random Effects Meta-Analysis of Correlations*

Description

Pools correlations across independent studies on the Fisher's Z scale and reports the results back in the correlation metric. Optionally, each study's correlation is first corrected for attenuation due to measurement error in either or both variables (the Spearman correction of correction_for_attenuation, the basic artifact correction of Hunter and Schmidt's psychometric meta-analysis), using reliabilities you supply, for example from the reliability family. That combination, synthesis connected to an actual reliability toolkit, is the measurement-aware path: the pooled quantity is then the construct-level correlation rather than the attenuated observed one.

Usage

```
meta_r(
  r,
  n,
  reliability_x = NULL,
  reliability_y = NULL,
  method = c("reml", "pm", "dl", "fe"),
  hartung_knapp = TRUE,
  conf_level = 0.95
)
```

Arguments

`r`	Numeric vector of observed correlations, one per study, each in (-1, 1).
`n`	Per-study sample sizes (integer, at least 4).
`reliability_x, reliability_y`	Optional per-study reliabilities in (0, 1] for the two measured variables; a single value is recycled across studies. When either is supplied, each correlation is disattenuated by $r_i/\sqrt{\rho_{xx,i}\,\rho_{yy,i}}$ before pooling (a reliability left `NULL` is treated as 1). The reliabilities are treated as known.
`method, hartung_knapp, conf_level`	Passed to `meta_es`.

Details

Pooling uses $z_i = \operatorname{atanh}(r_i)$ with sampling variance $1/(n_i - 3)$; the pooled estimate, its confidence limits, and the prediction interval are transformed back through $\tanh$. The heterogeneity quantities (tau, tau-squared, I-squared, H-squared, Q) remain on the Fisher's *Z* scale, where the model lives; tau is therefore the between-study standard deviation of the z-scale correlations.

When corrections are applied, the corrected correlation's variance is computed from its own n_i on the z scale, the conventional simple treatment when reliabilities are taken as known constants; the more elaborate artifact-distribution machinery of Hunter and Schmidt (2004) is deliberately out of scope here. A corrected correlation that exceeds 1 in magnitude (possible when an observed r outruns the supplied reliabilities) is an error at the pooling stage, unlike the single-study `correction_for_attenuation`, which reports it with a warning: atanh is undefined there.

Value

A `data.frame` (class `dmar_tbl`) with the same rows as `meta_es`: the `estimate`, `lower_limit` / `upper_limit`, and prediction interval rows in the correlation metric; the `se`, test statistic, and heterogeneity rows on the Fisher's *Z* scale where the model lives.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Hunter, J. E., & Schmidt, F. L. (2004). *Methods of meta-analysis: Correcting error and bias in research findings* (2nd ed.). Sage.

See Also

`meta_es` for the engine; `correction_for_attenuation` for the single-study correction and its connection to latent variable modeling; `reliability` for estimating the reliabilities; `convert_r_Z` / `convert_Z_r` for the transformation used.

Other meta-analysis: `combine_p()`, `meta_contrast()`, `meta_es()`, `meta_smd()`, `plot_forest()`

Examples

```
# Five validity studies of the same selection instrument.
r <- c(.28, .35, .22, .40, .31)
n <- c(120, 85, 200, 60, 150)
meta_r(r, n)

# The same studies corrected for criterion unreliability (reliability
# 0.80 in every study): the construct-level validity.
meta_r(r, n, reliability_y = 0.80)
```

---

meta_smd *Random Effects Meta-Analysis of Standardized Mean Differences*

Description

Pools two-group standardized mean differences across independent studies. Each study contributes its standardized mean difference and per-group sample sizes; the function computes the within-study sampling variances, applies the Hedges small-sample bias correction by default (the same J factor as `expected_smd` and `smd`), and fits the random effects model of `meta_es`, returning the pooled effect with its confidence interval, tau and tau-squared with intervals, I-squared, Cochran's Q, and a prediction interval for the effect in a new study.

Usage

```
meta_smd(
  smd,
  n_1,
  n_2,
  unbiased = TRUE,
  method = c("reml", "pm", "dl", "fe"),
  hartung_knapp = TRUE,
  conf_level = 0.95
)
```

Arguments

`smd`	Numeric vector of standardized mean differences (Cohen's d), one per study, positive in the direction of the common hypothesis.
`n_1`, `n_2`	Per-group sample sizes for each study.
`unbiased`	Logical: convert each d to Hedges g (the small-sample unbiased estimator) before pooling? Default `TRUE`. Set `FALSE` to pool the raw d values, for example when reproducing a historical analysis such as Raudenbush (1984) that predates routine use of the correction.
`method`, `hartung_knapp`, `conf_level`	Passed to `meta_es`: the tau-squared estimator (`"reml"` default), the Hartung-Knapp small-sample adjustment (default `TRUE`), and the confidence level.

Details

The within-study variance is the standard large-sample form

$$v_i = \frac{n_{1i} + n_{2i}}{n_{1i} n_{2i}} + \frac{g_i^2}{2(n_{1i} + n_{2i})},$$

computed from the bias-corrected g_i when `unbiased = TRUE` (Hedges, 1981; Borenstein, Hedges, Higgins, & Rothstein, 2009). All reported quantities are in the standardized mean difference metric.

Value

A `data.frame` (class `dmar_tbl`) with the same rows as `meta_es`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Borenstein, M., Hedges, L. V., Higgins, J. P. T., & Rothstein, H. R. (2009). *Introduction to meta-analysis*. Wiley.

Hedges, L. V. (1981). Distribution theory for Glass's estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Raudenbush, S. W. (1984). Magnitude of teacher expectancy effects on pupil IQ as a function of the credibility of expectancy induction: A synthesis of findings from 18 experiments. *Journal of Educational Psychology, 76*(1), 85–97.

See Also

`meta_es` for the engine and the reported rows; `smd` and `ci_smd` for the single-study quantities; `plot_forest` for the picture; `teacher_expectancy` for the example data.

Other meta-analysis: `combine_p()`, `meta_contrast()`, `meta_es()`, `meta_r()`, `plot_forest()`

Examples

```
# Pool the teacher expectancy studies (Raudenbush, 1984). Hedges g and
# the Hartung-Knapp adjustment are on by default; the prediction
# interval shows where a new expectancy study would be expected to land.
data(teacher_expectancy)
meta_smd(smd = teacher_expectancy$d,
         n_1 = teacher_expectancy$n_experimental,
         n_2 = teacher_expectancy$n_control)
```

`mixed_anova` *Mixed-Model ANOVA F-Ratios for One- and Two-Way Designs*

Description

Computes the classical *F*-ratios for one- and two-way ANOVA designs when one or more factors are random rather than fixed, using the expected-mean-square (EMS) rules that determine the correct denominator for each test (Searle, Casella, & McCulloch, 1992). The function is the closed-form alternative to fitting via `lmer` and is the standard treatment in classical psychometrics and design-of-experiments texts (Maxwell, Delaney, & Kelley, 2027, Ch. 10).

Usage

```
mixed_anova(
  data,
  outcome,
  factor_A,
  factor_B = NULL,
  A_type = c("fixed", "random"),
  B_type = c("random", "fixed")
)
```

Arguments

`data`	A `data.frame` containing the response, the factor(s), and (when both factors are crossed) the subject identifier.
`outcome`	Character name of the response column.
`factor_A`	Character name of factor *A*.
`factor_B`	Character name of factor *B*, or `NULL` for one-way designs. Default `NULL`.
`A_type`	One of `"fixed"` (default) or `"random"`: the EMS classification of factor *A*.
`B_type`	One of `"fixed"` or `"random"` (default): the EMS classification of factor *B*. Ignored when `factor_B` is `NULL`.

Details

One way design (only `factor_A`).

- Both *A* fixed and *A* random use the same observed *F*-ratio MS_A/MS_{within}; the test of "is there an effect of *A*?" is identical. The interpretation differs: the random-effects test asks whether the variance component σ^2_A is zero.

Two-way design, both fixed (Model I).

- $F_A = MS_A/MS_{AB}$ (when interaction is present in the model and treated as error) or $F_A = MS_A/MS_{\text{within}}$ (when interaction is pooled into error). The function uses MS_{within} as the denominator throughout for Model I.

Two-way design, both random (Model II).

- $F_A = MS_A/MS_{AB}$, $F_B = MS_B/MS_{AB}$, $F_{AB} = MS_{AB}/MS_{\text{within}}$.

Two-way mixed design (Model III, e.g., *A* fixed, *B* random).

- $F_A = MS_A/MS_{AB}$ (fixed factor against the interaction with the random factor)
- $F_B = MS_B/MS_{\text{within}}$ (random factor against the within-cell residual)
- $F_{AB} = MS_{AB}/MS_{\text{within}}$.

Balanced data assumed. The classical EMS rules require equal cell sizes. The function errors out on unbalanced data and recommends a mixed-effects fit via `lmer`.

Sums of squares. For the balanced designs this function targets, the Type I, Type II, and Type III sums of squares for each effect coincide, so the decomposition is unambiguous and no sums-of-squares type needs to be chosen (Maxwell, Delaney, & Kelley, 2027, Ch. 7). The returned object carries a numeric `sum_of_squares_type` attribute equal to 3, with the understanding that it equals Types I and II here; it records the convention without implying a choice that would matter for these designs.

Value

A `data.frame` with one row per testable effect. Columns: `effect`, `ss`, `df`, `ms`, `denominator`, `F_value`, `p_value`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 7 on the sums of squares for balanced designs and Chapter 10 on random and mixed effects.)

Searle, S. R., Casella, G., & McCulloch, C. E. (1992). *Variance components*. Wiley.

See Also

`anova_within`, `anova_within_two_way`, `lmer`

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `correlations_test()`, `equivalence_r()`, `equivalence_smd()`, `factorial_anova()`, `manova_split_plot()`, `mauchly_test()`, `obrien_test()`, `pairwise_within()`, `randomization_test()`, `randomization_test_paired()`, `regions_of_significance()`, `simple_effects_AB()`, `summary_t_test()`, `welch_t()`

Other mixed models: `R2_mixed_effects()`, `R2_mixed_effects_decomposition()`, `icc_lmer()`, `manova_split_plot()`, `ss_aipe_mixed_effects()`, `ss_aipe_mixed_effects_sensitivity()`, `ss_power_mixed_effects()`, `ss_power_split_plot_anova()`

Examples

```
# 1. Two-way mixed design: A fixed, B random.
set.seed(113)
grid <- expand.grid(A = factor(1:3), B = factor(1:5), rep = 1:4)
grid$y <- with(grid, 0.8 * as.integer(A) + rep(rnorm(5, 0, 1.5), each = 1)[as.integer(B)] +
                     rnorm(nrow(grid), 0, 1))
mixed_anova(grid, outcome = "y", factor_A = "A", factor_B = "B",
            A_type = "fixed", B_type = "random")
```

`mlmr` *Maximum Likelihood Multiple Regression*

Description

Fits a multiple regression model by maximum likelihood, with full information likelihood handling of missing values by default. The formula interface and S3 methods mirror `lm` so that calls such as `coef()`, `vcov()`, `confint()`, `summary()`, `fitted()`, `residuals()`, and `predict()` continue to work. Confidence intervals default to the likelihood ratio (profile) form; Wald and bootstrap variants are also available.

Usage

```
mlmr(
  formula,
  data,
  missing = c("fiml", "ml", "listwise", "pairwise", "available.cases"),
  ci_method = c("profile", "wald", "boot"),
  conf_level = 0.95,
  B = 1000L,
  boot_type = c("ordinary", "bollen.stine"),
  boot_seed = NULL,
  estimator = c("ML", "MLR", "MLM", "GLS"),
  se = NULL,
  fixed_x = FALSE,
  auxiliary = NULL,
  effect_sizes = TRUE,
  enforce_es_bounds = FALSE,
  ...
)
```

Arguments

`formula`	A two-sided `formula` of the form `y ~ x1 + x2 + ...`. Factor predictors, interactions (`x1 * x2`), polynomial terms (`poly(x, 2)`), and transformations (`I(x^2)`) are supported through the usual `model.matrix` expansion. An intercept-only formula, `y ~ 1`, fits the null model (the mean and the residual variance only); it

is the natural restricted model in a model comparison and pairs with `anova()` for a likelihood ratio test against a fuller model.

`data` A `data.frame` containing the variables in `formula`. Rows with missing values on *any* of the modeled variables are retained (under `missing = "fiml"`) and contribute to the likelihood through whichever components are observed.

`missing` Character; how missing values are handled. The default `"fiml"` (equivalently `"ml"` in **lavaan**) uses the full information maximum likelihood over all rows that have at least one observed value on the modeled variables. Other choices are `"listwise"` (drop rows with any missing modeled variable), `"pairwise"` (sample moments computed pairwise; not recommended with this model), and `"available.cases"`.

`ci_method` Character; the method for confidence intervals on the regression coefficients. `"profile"` (default) inverts the likelihood ratio test on each parameter through a sequence of constrained refits. `"wald"` returns the symmetric estimate $\pm\, z_{1-\alpha/2}$ standard error interval. `"boot"` resamples the rows of `data` `B` times, refits the model on each resample, and reports the percentile interval of the resampled slopes.

`conf_level` Desired level of confidence (the complement of the Type I error rate). Defaults to `0.95`.

`B` Integer; number of bootstrap resamples when `ci_method = "boot"`. Defaults to `1000`.

`boot_type` Character; `"ordinary"` (default) for the nonparametric resampling of rows, or `"bollen.stine"` for the Bollen and Stine (1992) model-based bootstrap implemented in **lavaan**.

`boot_seed` Integer or `NULL`; optional seed for the bootstrap resampling RNG. The default `NULL` leaves the user's current RNG state untouched, so successive bootstrap calls draw fresh resamples; supply an integer to make the bootstrap reproducible. When supplied, the function seeds the RNG locally and restores the prior state on exit so the user's global RNG is not polluted.

`estimator` Character; the **lavaan** estimator. One of `"ML"` (default), `"MLR"`, `"MLM"`, or `"GLS"`. `ML` is standard maximum likelihood under conditional normality of *Y*. `MLR` is robust maximum likelihood with Yuan-Bentler scaled standard errors and a Yuan-Bentler scaled test statistic, recommended when the conditional distribution of *Y* departs from normality and FIML is in use (Yuan & Bentler, 2000). `MLM` is Satorra- Bentler scaled χ^2 statistics under complete data (Satorra & Bentler, 1994). `GLS` is generalized least squares, an alternative ML-family estimator that is less commonly used in modern practice. The ordinal-data estimators (`"DWLS"`, `"WLS"`, `"ULS"` and their robust variants) are intentionally not exposed; for ordinal outcomes, fit a different model class.

`se` Character or `NULL`; the standard error type passed to **lavaan**. When `NULL` (default), the standard error type is chosen automatically from `estimator`: `"ML"` and `"GLS"` use `"standard"`, `"MLR"` uses `"robust.huber.white"` (Huber-White heteroskedasticity consistent), and `"MLM"` uses `"robust.sem"` (Satorra-Bentler). Override only when the default does not match the desired analysis. Other values include `"robust"`, `"first.order"`, and `"none"`.

`fixed_x` Logical; whether to treat predictors as fixed (not modeled jointly) or as random (jointly modeled). Defaults to `FALSE`, which is required for the full information likelihood to use rows with missing predictors. Set to `TRUE` only when the predictors are fully observed and the user wants the `lm`-style conditional model.

`auxiliary` Character vector of variable names in `data` to include as auxiliary variables, or `NULL` (default) for none. Auxiliaries are entered as saturated correlates (Graham, 2003): correlated with the outcome residual, every predictor, and each other, but not as predictors, so the regression coefficients keep their meaning while the full information maximum likelihood draws on the auxiliaries' observed values (the inclusive analysis strategy; Collins, Schafer, & Kam, 2001). A name in `auxiliary` must be numeric, must be present in `data`, must not appear in `formula`, and requires `fixed_x = FALSE`.

`effect_sizes` Logical; whether to compute regression effect sizes (standardized betas, semi-partial R^2, Cohen's f^2 per predictor, and the overall LR omnibus test). Defaults to `TRUE`. Disabling saves $K + 1$ additional lavaan refits.

`enforce_es_bounds`
Logical; whether to clamp semi-partial R^2 and Cohen's f^2 estimates to their theoretical lower bound of zero. Defaults to `FALSE`: the raw maximum likelihood estimates of R^2_{reduced} and R^2_{full} are reported as-is, and the difference can be slightly negative as a finite-sample artifact when the two are nearly equal. Setting to `TRUE` replaces any negative value with zero, which yields an estimate that respects the parameter space but is no longer the maximum likelihood estimate. When the clamp fires, the affected rows of the returned `effect_sizes` table carry the attribute `"clamped"` for diagnostics.

`...` Additional arguments forwarded to `lavaan`.

Details

Why a separate function from `lm`. `lm` uses ordinary least squares and listwise deletes any row with a missing value on the outcome or on any predictor. Two situations motivate a maximum likelihood alternative.

First, when a predictor is missing on some rows, listwise deletion can be biased if the missingness mechanism depends on other observed variables (the missing at random or MAR pattern). Full information maximum likelihood (FIML) jointly models the distribution of $(Y, X_1, \ldots, X_K)$ and yields consistent regression estimates under MAR, while listwise estimates can be biased away from the population values (Enders, 2010; Schafer & Graham, 2002).

Second, the joint likelihood estimates the predictor distribution as well, so quantities that depend on the predictor moments (the standardized coefficients, the model implied R^2, and the predictor variances and covariances) draw on every row with an observed predictor, not only the rows that are complete on the outcome. When the missing values are confined to the outcome, however, the unstandardized slopes and their standard errors match listwise deletion up to the maximum likelihood N versus $N-K-1$ variance divisor. The rows with an observed predictor but a missing outcome inform the marginal distribution of *X*, not the conditional distribution of *Y* given *X* that identifies the slopes, so they leave the slope estimates and their conditional-model standard errors unchanged.

The full information advantage is largest when (i) any predictors are missing on some rows, (ii) auxiliary variables that correlate with the outcome or with the missingness mechanism are supplied

through auxiliary (see below), or (iii) the bootstrap is used to obtain inference that does not depend on the multivariate normality assumption.

Auxiliary variables. A variable that is not part of the regression but is correlated with the outcome or with the missingness can be supplied through auxiliary. Auxiliaries are added to the model as saturated correlates (Graham, 2003): each one is correlated with the residual of the outcome, with every predictor, and with every other auxiliary, but is never entered as a predictor. The focal regression coefficients keep their meaning (on complete data they are unchanged to working precision), while the full information maximum likelihood uses the auxiliaries' observed values to make the MAR assumption hold conditional on more of the observed data and to recover information that listwise deletion discards. This is the inclusive analysis strategy of Collins, Schafer, and Kam (2001): a variable that predicts the missingness or the incomplete outcome belongs in the analysis even when it is of no substantive interest. Auxiliary variables must be numeric and require fixed_x = FALSE (the default).

Why likelihood ratio confidence intervals by default. Wald intervals (point estimate $\pm$ $z_{1-\alpha/2}$ standard error) are symmetric by construction and assume the sampling distribution of the estimator is approximately normal over the relevant range. Likelihood ratio intervals invert the likelihood ratio test directly: an interval contains every value of the parameter that would not be rejected at level α. Likelihood ratio intervals are invariant under monotone reparameterizations, often have better coverage in small samples, and respect parameter boundaries (Pawitan, 2001). The cost is computational: each parameter requires a sequence of refits with that parameter constrained. The examples below ask for the Wald and bootstrap intervals so the help page stays quick; a reported analysis leaves ci_method at its default.

The bootstrap interval. With ci_method = "boot" the rows of data are resampled with replacement B times (1000 by default) and the model is refit on each resample; with boot_type = "bollen.stine" the resamples are instead drawn from data transformed to satisfy the fitted model (Bollen & Stine, 1992), a model-based bootstrap. Only the percentile interval is offered: each coefficient's limits are the empirical quantiles of its resampled estimates (Efron & Tibshirani, 1993); there is no bias-corrected and accelerated (BCa) or bootstrap standard error variant. Resamples on which the refit does not converge are dropped, and the interval is computed from the resamples that return a value. The default B = 1000 is adequate for the central quantiles a percentile interval uses; raising it tightens the Monte Carlo error of the reported limits. Bootstrap results vary from run to run; supply boot_seed for reproducibility.

Model representation. Internally the model is fit through **lavaan** as a structural equation model in which *Y* is regressed on the predictors and (when fixed_x = FALSE) the predictor distribution is also estimated. With fixed_x = FALSE and complete data, point estimates of the slopes are identical to lm and the residual variance differs only by the usual N versus $N - K - 1$ divisor.

Caveats. The function assumes that, conditional on the modeled predictors, the dependent variable is normally distributed with constant variance. Missingness is assumed to be at most MAR; missing not at random patterns require selection or pattern mixture models outside the scope of this function. Factor predictors and interactions are expanded through model.matrix and entered as numeric covariates, so the same caveats about dummy variable encoding that apply to lm apply here as well.

Value

An object of class "mlmr", a list with components modeled on the structure of an lm fit:

call The matched call.

`formula` The model formula.

`terms` The terms object.

`model` The model frame (with missing values preserved when `missing = "fiml"`).

`coefficients` Named numeric vector of regression coefficients, with `(Intercept)` first when an intercept is in the formula.

`vcov` The variance-covariance matrix of the regression coefficients, returned by `vcov()`.

`ci` A two-column matrix (`lower`, `upper`) of confidence limits in the order of `coefficients`.

`ci_method` Which method was used to compute `ci`.

`conf_level` The confidence level used.

`coef_table` A `data.frame` with columns `term`, `estimate`, `se`, `z_value`, `p_value`, `ci_lower`, `ci_upper`.

`sigma2` Residual variance of *Y*, on the maximum likelihood scale (divisor *N*, not $N - K - 1$).

`R2` Model implied squared multiple correlation, $1 - \hat{\sigma}^2_e / \hat{\sigma}^2_Y$, where both variances come from the FIML estimated model implied covariance matrix.

`adj_R2` Adjusted R^2 using the number of complete cases (`N_complete`, the rows complete on the outcome and every predictor, which are the rows that identify the regression) and the number of slopes; the lavaan reported `N` can be larger under FIML because it counts rows that inform only the predictor distribution.

`logLik` The log likelihood at the maximum, with attributes `df` and `nobs` for compatibility with `stats::AIC` and `stats::BIC`.

`N` Sample size used by lavaan (rows with at least one observed value when `missing = "fiml"`; rows with no missing values when `missing = "listwise"`).

`N_complete` Number of rows that are complete on all modeled variables.

`fitted.values` Vector of fitted values, length `nrow(data)`, with `NA` for rows missing any predictor.

`residuals` Vector of residuals (`y - fitted`), with `NA` where `y` or any predictor was missing.

`lavaan_fit` The underlying **lavaan** fit object, returned for advanced users who want to apply **lavaan** accessors directly.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bollen, K. A., & Stine, R. A. (1992). Bootstrapping goodness of fit measures in structural equation models. *Sociological Methods & Research, 21*, 205–229. doi:10.1177/0049124192021002004

Collins, L. M., Schafer, J. L., & Kam, C.-M. (2001). A comparison of inclusive and restrictive strategies in modern missing data procedures. *Psychological Methods, 6*(4), 330–351. doi:10.1037/1082989X.6.4.330

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, NY: Chapman & Hall/CRC.

Enders, C. K. (2010). *Applied missing data analysis*. New York, NY: Guilford Press.

Graham, J. W. (2003). Adding missing-data-relevant variables to FIML-based structural equation models. *Structural Equation Modeling, 10*(1), 80–100. doi:10.1207/S15328007SEM1001_4

Pawitan, Y. (2001). *In all likelihood: Statistical modelling and inference using likelihood*. Oxford, UK: Oxford University Press.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48*(2), 1–36. doi:10.18637/jss.v048.i02

Satorra, A., & Bentler, P. M. (1994). Corrections to test statistics and standard errors in covariance structure analysis. In A. von Eye & C. C. Clogg (Eds.), *Latent variables analysis: Applications for developmental research* (pp. 399–419). Sage.

Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods, 7*, 147–177. doi:10.1037/1082989X.7.2.147

Yuan, K.-H., & Bentler, P. M. (2000). Three likelihood-based methods for mean and covariance structure analysis with nonnormal missing data. *Sociological Methodology, 30*(1), 165–200. doi:10.1111/00811750.00078

See Also

lm, sem, lavaan, ci_reg_coef, ci_rc, ci_src.

Examples

```
# Complete data: the maximum likelihood estimates agree with lm() to
# working precision. The fit asks for the Wald interval; the choice
# among the intervals is taken up below.
fit_mlmr <- mlmr(t6_paragraph_comprehension ~ t5_general_information +
                   t9_word_meaning,
                 data = holzinger_swineford, ci_method = "wald")
fit_lm   <- lm(t6_paragraph_comprehension ~ t5_general_information +
                 t9_word_meaning,
               data = holzinger_swineford)
cbind(mlmr = coef(fit_mlmr), lm = coef(fit_lm))

fit_mlmr
confint(fit_mlmr)

# summary() adds the intervals, the per-predictor semi-partial R^2
# and Cohen's f^2, and the omnibus likelihood ratio test of all
# slopes equal to zero.
summary(fit_mlmr)

# The interval menu is profile, Wald, and bootstrap. The default,
# ci_method = "profile", inverts the likelihood ratio test one
# parameter at a time through a sequence of constrained refits; it
# is what a reported interval deserves, and leaving ci_method at its
# default asks for it. The fits on this page ask for the Wald or the
# bootstrap interval because those refits take longer than a help
# page should. The bootstrap resamples rows and takes percentile
# limits; it is what to ask for when the normality the likelihood
# assumes is doubtful. B = 20 keeps the example quick, since every
# resample refits the model; a reported interval deserves the
```

```
# default B = 1000. boot_seed fixes the resamples, so the limits
# are reproducible rather than moving from run to run, and
# effect_sizes = FALSE skips the effect size refits, which
# summary() above already showed.
fit_boot <- mlmr(t6_paragraph_comprehension ~ t5_general_information +
                   t9_word_meaning, data = holzinger_swineford,
                 ci_method = "boot", B = 20, boot_seed = 113,
                 effect_sizes = FALSE)
confint(fit_boot)

# Missing values on a predictor are where maximum likelihood and
# least squares part company. The full information likelihood keeps
# every row that carries information; listwise deletion keeps only
# the rows that are complete. The Holzinger and Swineford battery
# carries real missingness for this: the revised second-form test
# t26_flags was administered to only 145 of the 301 students, so a
# model using it loses more than half the sample under listwise
# deletion while the full information fit keeps all 301 rows. Notice
# the two sample sizes, and that the two fits do not agree on the
# intercept.
fit_fiml <- mlmr(t6_paragraph_comprehension ~ t7_sentence +
                   t26_flags, data = holzinger_swineford,
                 ci_method = "wald", effect_sizes = FALSE)
fit_lwd  <- mlmr(t6_paragraph_comprehension ~ t7_sentence +
                   t26_flags, data = holzinger_swineford,
                 missing = "listwise",
                 ci_method = "wald", effect_sizes = FALSE)
rbind(FIML = coef(fit_fiml), listwise = coef(fit_lwd))
c(N_fiml = nobs(fit_fiml), N_listwise = nobs(fit_lwd))

# An auxiliary variable is not a predictor. The complete speed test
# t13_straight_and_curved_capitals enters as a saturated correlate,
# correlated with the outcome residual and with the predictors, so
# the likelihood can draw on it for the rows where t26_flags is
# missing while the coefficients keep their meaning. Continuing from
# the model above, the coefficients move a little and the slope
# standard errors shrink: the information the auxiliary recovers
# shows mostly as precision.
fit_aux <- mlmr(t6_paragraph_comprehension ~ t7_sentence +
                  t26_flags, data = holzinger_swineford,
                ci_method = "wald",
                auxiliary = "t13_straight_and_curved_capitals",
                effect_sizes = FALSE)
cbind(no_aux = coef(fit_fiml), aux = coef(fit_aux))
cbind(se_no_aux = sqrt(diag(vcov(fit_fiml))),
      se_aux = sqrt(diag(vcov(fit_aux))))
```

---

mlmr_mv *Multivariate Maximum Likelihood Regression With Full Information Missing Data Handling*

Description

Fits a multivariate multiple regression model by maximum likelihood, with full information maximum likelihood handling of missing values by default. Multiple outcomes are regressed on a shared predictor set simultaneously, with the residual covariance among outcomes estimated as part of the model. The formula interface mirrors `lm`'s multivariate syntax, `cbind(y1, y2, y3) ~ x1 + x2`, and the returned object supports the same family of S3 methods as a univariate `mlmr` fit.

Usage

```
mlmr_mv(
  formula,
  data,
  missing = c("fiml", "ml", "listwise", "pairwise", "available.cases"),
  ci_method = c("profile", "wald", "boot"),
  conf_level = 0.95,
  B = 1000L,
  boot_type = c("ordinary", "bollen.stine"),
  boot_seed = NULL,
  estimator = c("ML", "MLR", "MLM", "GLS"),
  se = NULL,
  fixed_x = FALSE,
  auxiliary = NULL,
  effect_sizes = TRUE,
  enforce_es_bounds = FALSE,
  ...
)
```

Arguments

`formula`	A two-sided `formula` whose left-hand side is `cbind(y1, y2, ...)` for two or more numeric outcomes and whose right-hand side names the shared predictor set. Factor predictors, interactions, polynomial terms, and transformations are supported as in `mlmr`.
`data`	A `data.frame` containing the variables in `formula`.
`missing`	Character; passed to **lavaan**. Defaults to `"fiml"` (equivalently `"ml"`). See `mlmr` for the full list of accepted values.
`ci_method`	Character; confidence interval method for the regression coefficients. `"profile"` (default), `"wald"`, or `"boot"`. The same trade-offs apply as in `mlmr`; with J outcomes and K predictors, profile likelihood requires $O(JK)$ constrained refits. `"boot"` resamples the rows of `data` `B` times (or draws Bollen-Stine model-based resamples), refits on each, drops resamples whose refit does not converge, and reports each coefficient's percentile interval, the empirical quantiles of its resampled estimates; as in `mlmr`, the percentile interval is the only bootstrap interval offered. The examples ask for the Wald and bootstrap intervals so the help page stays quick; a reported analysis leaves `ci_method` at its default.
`conf_level`	Desired level of confidence. Defaults to `0.95`.

B	Integer; number of bootstrap resamples when ci_method = "boot". Defaults to 1000.
boot_type	Character; "ordinary" (default) or "bollen.stine".
boot_seed	Integer or NULL. Defaults to NULL, which leaves the user's current RNG state intact; supply an integer for reproducible bootstraps. When set, the seed applies for the duration of the call and the caller's generator state is restored on exit.
estimator	Character; one of "ML" (default), "MLR", "MLM", "GLS".
se	Character or NULL; defaults to NULL, meaning "choose automatically from estimator" (same logic as mlmr).
fixed_x	Logical; defaults to FALSE (jointly model the predictor distribution, required for FIML to use rows with missing predictors).
auxiliary	Character vector of variable names in data to include as auxiliary variables (saturated correlates; Graham, 2003), or NULL (default) for none. Each auxiliary is correlated with every outcome's residual, every predictor, and each other auxiliary, but is never entered as a predictor, so the per-outcome regression coefficients keep their meaning while the full information maximum likelihood draws on the auxiliaries' observed values (the inclusive analysis strategy; Collins, Schafer, & Kam, 2001). A name in auxiliary must be numeric, present in data, absent from formula, and requires fixed_x = FALSE.
effect_sizes	Logical; whether to compute per-outcome standardized betas, semi-partial R^2, and Cohen's f^2. Defaults to TRUE.
enforce_es_bounds	Logical; if TRUE, negative per-outcome sr^2 or f^2 estimates (finite-sample artifacts) are clamped to zero. Defaults to FALSE.
...	Additional arguments forwarded to lavaan.

Details

Why a separate function from mlmr. A univariate mlmr fit handles the case of one outcome regressed on one or more predictors. mlmr_mv extends to the case of two or more outcomes regressed on the same predictor set, modeling the residual covariance among outcomes explicitly. This is the regression problem in which the FIML advantage over listwise deletion is largest, because rows that are missing on one outcome still contribute information about the other outcomes (via the modeled residual covariance) and about the joint distribution of the predictors. A user who fits separate univariate regressions for each outcome under listwise deletion can discard a great deal of information when the outcomes are correlated and missingness patterns differ.

Same predictor set across outcomes. The formula cbind(y1, y2) ~ x1 + x2 regresses both y1 and y2 on x1 and x2. Per-outcome predictor sets (sometimes called seemingly unrelated regression with heterogeneous predictors) are not supported; users who need that can fit separate mlmr models or call sem directly with a custom model string.

Auxiliary variables. As in mlmr, variables that are not part of the regression but are correlated with an outcome or with the missingness can be supplied through auxiliary and are entered as saturated correlates (Graham, 2003): each is correlated with every outcome's residual, every predictor, and each other auxiliary, but never as a predictor, so the per-outcome coefficients keep their meaning while the likelihood draws on the auxiliaries' observed values (the inclusive analysis strategy of Collins, Schafer, & Kam, 2001).

The bootstrap interval. `ci_method = "boot"` resamples the rows of `data` with replacement `B` times (1000 by default), refits the model on each resample, and reports each coefficient's percentile interval. It is the interval to ask for when the multivariate normality the likelihood assumes is doubtful, since its coverage does not rest on that assumption. The price is `B` refits of a model that already carries J outcomes, so a bootstrap interval is a deliberate request rather than a default. Bootstrap results vary from run to run; supply `boot_seed` for reproducibility. The mechanics, including the Bollen-Stine variant, are given in the `ci_method` argument description and in `mlmr`.

Caveats. Same as `mlmr`: the function assumes that, conditional on the predictors, the joint distribution of the outcomes is multivariate normal with constant covariance, and that missingness is at most MAR. Factor predictors and interactions are expanded through `model.matrix` once and reused for every outcome.

Value

An object of class `"mlmr_mv"`, a list with components similar to a univariate `mlmr` fit but extended for multiple outcomes:

`call`, `formula`, `terms`, `model`, `xlevels` As in `mlmr`.

`coefficients` A matrix with predictors (and an intercept row, when present) as rows and outcomes as columns, matching `coef.mlm`.

`coef_table` A long `data.frame` with one row per (outcome, term) combination; columns include `outcome`, `term`, `estimate`, `se`, `z_value`, `p_value`, `ci_lower`, `ci_upper`, `std_estimate`.

`vcov` The variance-covariance matrix of the regression coefficients across all outcomes, returned by `vcov()`. Rows and columns follow the outcome-major order of `coef_table` (the column-major flattening of `coefficients`) and are named `"outcome:term"`, for example `"mpg:wt"`, the naming `vcov` uses for an `"mlm"` fit. The cross-outcome blocks carry the sampling covariance between coefficients of different outcomes, so joint Wald tests across outcomes compose with `coef()`.

`residual_cov` The estimated residual covariance matrix among outcomes (J by J).

`R2` Named vector of model implied R^2 per outcome.

`adj_R2` Named vector of adjusted R^2 per outcome.

`effect_sizes` When `effect_sizes = TRUE`, a long `data.frame` with one row per (outcome, predictor) combination giving sr^2 and Cohen's f^2.

`fitted.values`, `residuals` Matrices with rows = observations and columns = outcomes; `NA` in rows where any predictor is missing.

`logLik`, `N`, `N_complete` As in `mlmr`.

`lavaan_fit` The underlying **lavaan** fit.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

`mlmr` for the univariate sibling; `lm` (and the `"mlm"` object class) for the OLS multivariate analog; `sem` for the underlying engine.

Examples

```
# Two outcomes on a shared predictor set. The residual covariance
# between the outcomes is estimated as part of the model, which is
# what separates this from two separate regressions. The fit asks
# for the Wald interval, a choice taken up below, and keeps the
# default effect_sizes = TRUE: the per-outcome effect sizes come
# back on the fit rather than in summary(), one row per outcome and
# predictor, giving the semi-partial R^2 and Cohen's f^2, and they
# are what fills the standardized coefficients in coef_table.
fit <- mlmr_mv(cbind(t6_paragraph_comprehension, t9_word_meaning) ~
                 t5_general_information + t7_sentence,
               data = holzinger_swineford,
               ci_method = "wald")
coef(fit)              # matrix: rows = predictors, cols = outcomes
summary(fit)
fit$R2                 # per-outcome R^2
fit$residual_cov       # residual covariance among outcomes
print(fit$effect_sizes, row.names = FALSE)

# The interval menu is profile, Wald, and bootstrap. The default,
# ci_method = "profile", inverts the likelihood ratio test one
# coefficient at a time, and with two outcomes there are twice as
# many coefficients to profile; it is what a reported interval
# deserves, and leaving ci_method at its default asks for it. The
# fits on this page ask for the Wald or the bootstrap interval
# because those refits take longer than a help page should. The
# bootstrap resamples rows and takes percentile limits; it is what
# to ask for when the multivariate normality the likelihood assumes
# is doubtful. B = 10 keeps the example quick, since every resample
# refits the two-outcome model; a reported interval deserves the
# default B = 1000. boot_seed fixes the resamples, so the limits
# are reproducible rather than moving from run to run. The effect
# sizes cost one constrained refit per outcome and predictor, so this
# fit and the ones after it leave them off.
fit_boot <- mlmr_mv(cbind(t6_paragraph_comprehension, t9_word_meaning) ~
                      t5_general_information + t7_sentence,
                    data = holzinger_swineford,
                    ci_method = "boot", B = 10, boot_seed = 113,
                    effect_sizes = FALSE)
confint(fit_boot)

# FIML versus listwise when one outcome has missing values. The
# revised second-form test t26_flags was administered to only 145
# of the 301 students, so it carries real missingness. A row with
# t26_flags missing still informs the likelihood about the other
# outcome, about the predictors, and, through the residual
# covariance, about t26_flags itself, so no row is discarded. Notice
# the two sample sizes, and that the coefficients of the complete
# outcome differ between the fits: listwise deletion drops 156 of
# its observed rows along with the missing t26_flags values.
fit_fiml <- mlmr_mv(cbind(t6_paragraph_comprehension,
                          t26_flags) ~
```

```
                      t7_sentence + t9_word_meaning,
                    data = holzinger_swineford,
                    ci_method = "wald", effect_sizes = FALSE)
fit_lwd  <- mlmr_mv(cbind(t6_paragraph_comprehension,
                          t26_flags) ~
                      t7_sentence + t9_word_meaning,
                    data = holzinger_swineford,
                    missing = "listwise", ci_method = "wald",
                    effect_sizes = FALSE)
c(N_fiml = nobs(fit_fiml), N_listwise = nobs(fit_lwd))
cbind(FIML = coef(fit_fiml)[, "t6_paragraph_comprehension"],
      listwise = coef(fit_lwd)[, "t6_paragraph_comprehension"])

# Auxiliary variable (saturated correlates): the complete speed test
# t13_straight_and_curved_capitals informs the likelihood without
# entering either regression. Continuing from the model above, the
# coefficients of the complete outcome are unchanged to working
# precision, while those of t26_flags move, since the auxiliary
# carries information about the rows where t26_flags is missing.
fit_aux <- mlmr_mv(cbind(t6_paragraph_comprehension,
                         t26_flags) ~
                     t7_sentence + t9_word_meaning,
                   data = holzinger_swineford,
                   ci_method = "wald",
                   auxiliary = "t13_straight_and_curved_capitals",
                   effect_sizes = FALSE)
coef(fit_aux)
```

`moments_nc_chisq` *Moments of the Noncentral Chi Square Distribution*

Description

Returns the mean, variance, standard deviation, skewness, and excess kurtosis of a noncentral chi square distribution with `df` degrees of freedom and noncentrality parameter `ncp`. The noncentral chi square is the distribution of a sum of squared independent normals with nonzero means ($\sum(Z_i + \mu_i)^2$, with $\lambda = \sum \mu_i^2$); it is the building block of the noncentral F (whose numerator is a noncentral chi square) and the reference distribution for likelihood ratio and Wald statistics under the alternative. Unlike the noncentral t and F, every moment exists, so none of the returned values is ever `NA`.

Usage

```
moments_nc_chisq(df, ncp = 0)
```

Arguments

`df`	Degrees of freedom, a single positive number (need not be a whole number).
`ncp`	Noncentrality parameter λ, a single non-negative number. Defaults to 0, the central chi square.

Details

The cumulants of the noncentral chi square are $\kappa_n = 2^{n-1}(n-1)!\,(\nu + n\lambda)$ for $n \geq 1$, from which the moments follow directly: the mean is $\kappa_1 = \nu + \lambda$, the variance is $\kappa_2 = 2(\nu + 2\lambda)$, the skewness is $\kappa_3/\kappa_2^{3/2} = \sqrt{8}\,(\nu + 3\lambda)/(\nu + 2\lambda)^{3/2}$, and the excess kurtosis is $\kappa_4/\kappa_2^2 = 12(\nu + 4\lambda)/(\nu + 2\lambda)^2$. At $\lambda = 0$ these reduce to the central chi square values: mean ν, variance 2ν, skewness $\sqrt{8/\nu}$, and excess kurtosis $12/\nu$.

Value

A `data.frame` (class `dmar_tbl`) in `term` / `value` layout with the `mean`, `variance`, `sd`, `skewness`, and `excess_kurtosis`, followed by the `df` and `ncp` that produced them.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous univariate distributions* (Vol. 2, 2nd ed., Chapter 29). Wiley.

See Also

`moments_nc_F` (whose numerator is a noncentral chi square) and `moments_nc_t` for the other noncentral moments; `ci_nc_chisq` for the noncentral chi square confidence limits; `dchisq` for the density.

Other noncentral distribution moments: `moments_nc_F()`, `moments_nc_t()`

Examples

```
# A noncentral chi square with 5 df and noncentrality 3.
moments_nc_chisq(df = 5, ncp = 3)

# ncp = 0 is the central chi square: mean df, variance 2 * df.
moments_nc_chisq(df = 5)

# Every moment exists for any positive df, so nothing is ever NA.
anyNA(moments_nc_chisq(df = 1, ncp = 10)$value)
```

`moments_nc_F` *Moments of the Noncentral* F *Distribution*

Description

Returns the mean, variance, standard deviation, skewness, and excess kurtosis of a noncentral *F* distribution with `df_1` numerator and `df_2` denominator degrees of freedom and noncentrality parameter `ncp`. The noncentral *F* is the reference distribution of the *F* statistic when an effect is present, so its moments describe the sampling behavior of R^2, eta squared, and the omnibus *F* test under the alternative. A central *F* (`ncp = 0`) is the special case. The function was `moments_ncf()` in earlier builds of DMAR.

Usage

```
moments_nc_F(df_1, df_2, ncp = 0)
```

Arguments

`df_1`	Numerator degrees of freedom, a single positive number.
`df_2`	Denominator degrees of freedom, a single positive number.
`ncp`	Noncentrality parameter λ, a single non-negative number. Defaults to 0, the central *F*.

Details

Writing the noncentral *F* as $F = (X_1/\nu_1)/(X_2/\nu_2)$ with $X_1 \sim \chi^2_{\nu_1}(\lambda)$ a noncentral chi square and $X_2 \sim \chi^2_{\nu_2}$ independent, the raw moments are

$$\mathrm{E}[F^k] = \left(\tfrac{\nu_2}{\nu_1}\right)^k \mathrm{E}[X_1^k] \prod_{i=1}^{k} \frac{1}{\nu_2 - 2i}, \qquad \nu_2 > 2k,$$

where the noncentral chi square moments $\mathrm{E}[X_1^k]$ follow from its cumulants $\kappa_n = 2^{n-1}(n-1)!\,(\nu_1 + n\lambda)$. The mean exists for $\nu_2 > 2$, the variance for $\nu_2 > 4$, the skewness for $\nu_2 > 6$, and the excess kurtosis for $\nu_2 > 8$; a moment whose denominator degrees of freedom condition is not met is returned as `NA`. The mean reduces to the familiar $\nu_2(\nu_1 + \lambda)/[\nu_1(\nu_2 - 2)]$, and the variance to $2(\nu_2/\nu_1)^2[(\nu_1 + \lambda)^2 + (\nu_1 + 2\lambda)(\nu_2 - 2)]/[(\nu_2 - 2)^2(\nu_2 - 4)]$.

Value

A `data.frame` (class `dmar_tbl`) in `term` / `value` layout with the `mean`, `variance`, `sd`, `skewness`, and `excess_kurtosis` (any of which may be `NA` when `df_2` is too small), followed by the `df_1`, `df_2`, and `ncp` that produced them.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous univariate distributions* (Vol. 2, 2nd ed., Chapter 30). Wiley.

See Also

moments_nc_t for the noncentral *t*; ci_nc_F for the noncentral *F* confidence limits used in effect size intervals; df for the density.

Other noncentral distribution moments: moments_nc_chisq(), moments_nc_t()

Examples

```
# A noncentral F with 3 and 40 df and noncentrality 8.
moments_nc_F(df_1 = 3, df_2 = 40, ncp = 8)

# ncp = 0 is the central F: mean df_2 / (df_2 - 2).
moments_nc_F(df_1 = 3, df_2 = 40)

# The variance is undefined for four or fewer denominator df.
moments_nc_F(df_1 = 2, df_2 = 4, ncp = 5)
```

---

moments_nc_t *Moments of the Noncentral* t *Distribution*

Description

Returns the mean, variance, standard deviation, skewness, and excess kurtosis of a noncentral *t* distribution with df degrees of freedom and noncentrality parameter ncp. These are the closed-form moments surveyed by Owen (1968); they are the engine behind the bias and variance of the standardized mean difference (Cohen's *d*), since *d* is a scaled noncentral *t* variate. A central *t* (ncp = 0) is the special case with mean 0 and the familiar $df/(df-2)$ variance. The function was moments_nct() in earlier builds of DMAR.

Usage

```
moments_nc_t(df, ncp = 0)
```

Arguments

df	Degrees of freedom, a single positive number (need not be a whole number).
ncp	Noncentrality parameter δ, a single number (may be negative, which mirrors the distribution about 0). Defaults to 0, the central *t*.

Details

Writing the noncentral *t* as $T = (Z+\delta)/\sqrt{W/\nu}$ with $Z \sim N(0,1)$ and $W \sim \chi^2_\nu$ independent, the raw moments are

$$\mathrm{E}[T^k] = \mathrm{E}[(Z+\delta)^k]\left(\tfrac{\nu}{2}\right)^{k/2}\frac{\Gamma\big((\nu-k)/2\big)}{\Gamma(\nu/2)}, \qquad \nu > k,$$

computed here on the log scale for stability. The mean exists for $\nu > 1$, the variance for $\nu > 2$, the skewness for $\nu > 3$, and the excess kurtosis for $\nu > 4$; a moment whose degrees of freedom condition is not met is returned as NA. The mean is $\delta\sqrt{\nu/2}\,\Gamma((\nu-1)/2)/\Gamma(\nu/2)$, the δ-scaled reciprocal of the Hedges (1981) bias-correction factor that `expected_smd` and `smd` use; that is why the standardized mean difference is upward biased.

Value

A `data.frame` (class `dmar_tbl`) in `term` / `value` layout with the `mean`, `variance`, `sd`, `skewness`, and `excess_kurtosis` (any of which may be NA when the degrees of freedom are too small), followed by the `df` and `ncp` that produced them.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Owen, D. B. (1968). A survey of properties and applications of the noncentral t-distribution. *Technometrics, 10*(3), 445–478. doi:10.1080/00401706.1968.10490590

Hedges, L. V. (1981). Distribution theory for Glass's estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

See Also

`moments_nc_F` for the noncentral *F*; `expected_smd` and `var_smd` for the same moments specialized to Cohen's *d*; `dt` for the density.

Other noncentral distribution moments: `moments_nc_F()`, `moments_nc_chisq()`

Examples

```
# A noncentral t with 20 df and noncentrality 2.5.
moments_nc_t(df = 20, ncp = 2.5)

# ncp = 0 is the central t: mean 0, variance df / (df - 2), no skew.
moments_nc_t(df = 10)

# The mean is the noncentrality times the Hedges bias factor's reciprocal,
# which is why Cohen's d (a scaled noncentral t) is upward biased.
m <- moments_nc_t(df = 18, ncp = 1.2)
m$value[m$term == "mean"]
```

mr_cv *Minimum Risk Point Estimation of the Population Coefficient of Variation*

Description

A function for the sequential estimation of the coefficient of variations with minimum risk. The function implements the ideas of Chattopadhyay and Kelley (2016), which considers study cost and accuracy of the estimated coefficient of variation simultaneously.

Usage

```
mr_cv(
  data,
  A,
  structural_cost,
  epsilon,
  sampling_cost,
  pilot = FALSE,
  m0 = 4,
  gamma = 0.49,
  verbose = FALSE
)
```

Arguments

data	the data for which to evaluate the function
A	$structural_cost/(epsilon^2)$; this is the structural cost that one is willing to pay in a study to estimate the coefficient of variation divided by the square of the desired difference (between the estimate and the parameter)
structural_cost	The structural cost of what one is willing to pay in a study (see note below)
epsilon	The maximum desired difference between the estimated coefficient of variation and the population value)
sampling_cost	The sampling cost to collect an additional observation. For example, if each survey costs 10 dollars to distribute and score, sampling_cost would be 10 dollars per additional observation
pilot	TRUE or FALSE based on whether the users is using the function to plan a pilot sample size (TRUE) or if it is being used to assess if the optimization criterion has been satisfied (FALSE)
m0	The minimum bound on the initial pilot sample size
gamma	A correction factor in which we suggest .49; see the two Chattopadhyay & Kelley articles for more details (ignorable for most users)
verbose	If TRUE, extra information is printed; defaults to FALSE

Details

The value of `epsilon` is context specific; the smaller the value the closer the estimated value will tend to be to the population value.

Value

`risk`	The value of the risk function
`n`	The current sample size
`cv`	The current coefficient of variation
`is_satisfied`	A TRUE/FALSE statement of whether or not the risk function has been satisfied. If TRUE then sampling can stop as the stopping rule has been satisfied

Note

When a study's aim is to estimate a parameter accurately, such as the coefficient of variation, the structural costs and the maximum probable error of the estimate (i.e., ϵ) are combined to form A. When we say "what the researcher is willing to pay", we literally mean the structural cost (c) the researcher is willing to invest in a study in order to estimate the parameter of interest with the desired degree of accuracy. This value is implicitly included (along with anticipated sampling cost) in grant applications for empirical studies when a certain amount of money is requested to conduct a study. If a researcher is willing to pay more and/or desire a smaller value of ϵ, A is larger than it would have been. A larger A value will translate into a more expensive study, holding everything else constant. Notice that A is a fixed value in any investigation, as the researcher specifies A directly or by specifying its two components (structural cost and ϵ) individually. However, what is not fixed but rather evaluated in multiple steps throughout the process is the sampling cost, as it is unknown the necessary sample size in order to accomplish the study's goal of achieving a sufficiently accurate estimate of the coefficient of variation. This is the core of our contributions: minimizing sampling cost, and thereby study cost, by using a sequential procedure that evaluates a stopping rule using the risk function to determine if the optimization criterion has been satisfied (based on the goals of the researcher and current information available). This function implements the ideas of sampling error and the study costs are considered simultaneously, so that the cost is not higher than necessary for the tolerable sampling error.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Chattopadhyay, B., & Kelley, K. (2016). Estimation of the coefficient of variation with minimum risk: A sequential method for minimizing sampling error and study cost. *Multivariate Behavioral Research, 51*(5), 627–648. doi:10.1080/00273171.2016.1203279

Chattopadhyay, B., & Kelley, K. (2017). Estimating the standardized mean difference with minimum risk: Maximizing accuracy and minimizing cost with sequential estimation. *Psychological Methods, 22*(1), 94–113. doi:10.1037/met0000089

Kelley, K. (2007). Sample size planning for the coefficient of variation from the accuracy in parameter estimation approach. *Behavior Research Methods, 39*(4), 755–766. doi:10.3758/BF03192966

Kelley, K., Darku, F. B., & Chattopadhyay, B. (2018). Accuracy in parameter estimation for a general class of effect sizes: A sequential approach. *Psychological Methods, 23*, 226–243. doi:10.1037/met0000127

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

ci_cv, cv, mr_smd

Examples

```
# Determine pilot sample size:
mr_cv(pilot = TRUE, A = 400000, sampling_cost = 75, gamma = .49)

# Collect data (the size of which is the pilot sample size)
Data <- c(36, 53, 19, 11, 10, 24, 14, 65, 18, 48, 25, 35, 13, 18, 3, 41, 5, 3)

# Use mr_cv() to assess if the criterion for stopping the sequential study has been satisfied:
mr_cv(data = Data, A = 400000, sampling_cost = 75, gamma = .49)

# Collect another data (m=1 here) and perform another check:
Data <- c(Data, 44)
mr_cv(data = Data, A = 400000, sampling_cost = 75, gamma = .49)

# Continue adding obervations, checking each time if m=1, until the minimum risk criteria
# are satisfied:
Data <- c(Data, 26, 13, 39, 2, 3, 26, 22, 8, 15, 12, 22, 5, 21, 23, 40, 18)
mr_cv(data = Data, A = 400000, sampling_cost = 75, gamma = .49)
```

---

mr_smd *Minimum Risk Point Estimation of the Population Standardized Mean Difference*

Description

A function for the sequential estimation of the standardized mean difference with minimum risk. The function implements the ideas of Chattopadhyay and Kelley (2017), which considers study cost and accuracy of the estimated standardized mean difference simultaneously. This is important to specify that mr_smd was developed under the assumption of normally distributed data with equal sample size and equal cost of sampling per observation for each group.

Usage

```
mr_smd(
  A,
  structural_cost,
```

```
  epsilon,
  d,
  n,
  sampling_cost,
  pilot = FALSE,
  m0 = 4,
  gamma = 0.49
)
```

Arguments

A	The price one is willing to pay in order to have a maximum allowable difference of $epsilon^2$ between the estimate of the standardized mean difference and its corresponding parameter
structural_cost	The structural cost of what one is willing to pay in a study
epsilon	The maximum desired difference between the estimated standardized mean difference and the population value
d	The current estimate of the standardized mean difference
n	Current sample size *per group* (thus total sample size is $2n$); requires equal sample size *per group*
sampling_cost	The sampling cost to collect an additional observation. For example, if each survey costs 10 dollars to distribute and score, `sampling_cost` would be 10 dollars per additional observation
pilot	`TRUE` or `FALSE` based on whether the users is using the function to plan a pilot sample size (TRUE) or if it is being used to assess if the optimization criterion has been satisfied (FALSE)
m0	The minimum bound on the initial pilot sample size
gamma	A correction factor in which we suggest .49; see the two Chattopadhyay & Kelley articles for more details (ignorable for most users)

Details

The standardized mean difference is a widely used measure effect size. In this article, we developed a general theory for estimating the population standardized mean difference by minimizing both the mean square error of the estimator and the total sampling cost. This function implements our ideas discussed in Chattopadhyay and Kelley (2017). See also Kelley and Rausch (2006) for additional information on the standardized mean difference.

Value

risk	The value of the risk function.
n1	Sample size for group 1 (echos the input value)
n2	Sample size for group 2 (echos the input value)
d	Observed value of the standardized mean difference (i.e., *d*; echos the input value)

`is_satisfied` A `TRUE` or `FALSE` statement of that evaluates a stopping rule using the risk function to determine if the optimization criterion has been satisfied (based on the goals of the researcher and current information available)

Note

When `pilot=TRUE` the function returns the size of the pilot sample size, *per group*, that should be used (thus, the total sample size is twice the pilot sample size).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Chattopadhyay, B., & Kelley, K. (2016). Estimation of the coefficient of variation with minimum risk: A sequential method for minimizing sampling error and study cost. *Multivariate Behavioral Research, 51*(5), 627–648. doi:10.1080/00273171.2016.1203279

Chattopadhyay, B., & Kelley, K. (2017). Estimating the standardized mean difference with minimum risk: Maximizing accuracy and minimizing cost with sequential estimation. *Psychological Methods, 22*(1), 94–113. doi:10.1037/met0000089

Kelley, K., Darku, F. B., & Chattopadhyay, B. (2018). Accuracy in parameter estimation for a general class of effect sizes: A sequential approach. *Psychological Methods, 23*, 226–243. doi:10.1037/met0000127

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

`ci_smd`, `mr_cv`

Examples

```
# To obtain pilot sample size in a situation in which A=10000. Note that 'A' is
# 'structural_cost' divided by the square of 'epsilon'.

# From Chattopadhyay and Kelley (2017)
mr_smd(pilot = TRUE, A = 10000, sampling_cost = 2.4, gamma = .49)

High.SLS <- c(11, 7, 22, 13, 6, 9, 11, 16, 12, 17, 14, 8, 16)
Low.SLS <- c(3, 6, 10, 8, 14, 5, 12, 10, 6, 8, 13, 5, 9)

mr_smd(d = 1.021484, n = 13, A = 10000, sampling_cost = 2.40, gamma = .49)

# Or, using the smd() function:
mr_smd(d = smd(group_1 = High.SLS, group_2 = Low.SLS)[1,2], n = 13, A = 10000,
```

```
        sampling_cost = 2.40, gamma = .49)

# Here, for this situation, the stopping rule is satisfied:
mr_smd(d = 1.00, n = 75, A = 10000, sampling_cost = 2.40, gamma = .49)
```

`nnt_from_smd` *Number Needed to Treat (NNT) From Cohen's* d

Description

Converts a standardized mean difference (Cohen's *d*) into the number needed to treat (NNT) using the Kraemer-Kupfer (2006) framework, which connects *d* to the success-rate difference (SRD) under the assumption of a continuous, normally distributed outcome with equal variances and a hypothetical median-cut criterion for treatment success. When a confidence interval on *d* is supplied (or noncentrality parameters / sample sizes are provided so that one can be constructed), the bounds are propagated through the same conversion to give a CI on the NNT.

Usage

```
nnt_from_smd(
  smd,
  n_1 = NULL,
  n_2 = NULL,
  conf_level = 0.95,
  smd_lower = NULL,
  smd_upper = NULL
)
```

Arguments

`smd`	Sample standardized mean difference (Cohen's *d*); a numeric scalar. Positive values correspond to the treatment group exceeding the control group.
`n_1`, `n_2`	Sample sizes in the treatment and control groups; both required when a noncentral *t*-based CI on the NNT is desired.
`conf_level`	Confidence level for the CI on the NNT (when `n_1` and `n_2` are supplied). Default `0.95`.
`smd_lower`, `smd_upper`	Optional pre-computed confidence limits on *d*. If supplied, these are used directly to propagate the interval through the SRD-to-NNT map and the noncentral *t* computation is skipped.

Details

The conversion. Under bivariate normality with equal variances, Kraemer & Kupfer (2006) showed that the proportion of times a randomly drawn treatment-group observation exceeds a randomly drawn control-group observation is

$$p \;=\; \Pr(Y_T > Y_C) \;=\; \Phi\big(d/\sqrt{2}\big),$$

from which the success-rate difference (their effect size) is

$$\mathrm{SRD} \;=\; 2p - 1 \;=\; 2\Phi\big(d/\sqrt{2}\big) - 1,$$

and the number needed to treat is its reciprocal,

$$\mathrm{NNT} \;=\; 1/\mathrm{SRD}.$$

Larger d produces smaller NNT; $d = 0$ produces $\mathrm{NNT} = \infty$ (no advantage). The conversion is monotone, so the SRD/NNT confidence interval is obtained by applying the same transformation to the endpoints of the CI on d; the lower NNT limit comes from the *upper* d limit and vice versa (Furukawa & Leucht, 2011).

When NNT becomes infinite or negative. If the lower CI on d is exactly zero, the corresponding upper NNT bound is `Inf`: the data do not exclude the possibility that the treatment produces no advantage (or even harm). Negative values of d are allowed; the function returns negative NNT values which are conventionally read as the NNT to *harm*.

Assumption check. The Kraemer-Kupfer conversion assumes a continuous, normally distributed outcome with equal variances across groups. For skewed outcomes, ordinal outcomes, or unequal variances, the empirical common-language effect size `cles` or the Vargha-Delaney A statistic is more defensible. Furukawa & Leucht (2011) compare four methods and recommend the Kraemer-Kupfer formula as the most accurate under normality.

Value

A `data.frame` with rows for the success-rate difference (`srd`), the point estimate of NNT (`nnt`), and (when an interval is constructable) the lower and upper NNT limits. When the lower CI on d is exactly zero, the corresponding NNT bound is reported as `Inf`; when it is negative, that bound is a finite negative value (the NNT to harm), so a CI on d that spans zero yields an NNT interval passing through the infinite point that separates benefit from harm.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Furukawa, T. A., & Leucht, S. (2011). How to obtain NNT from Cohen's d: Comparison of four methods. *PLoS ONE, 6*(4), e19070. doi:10.1371/journal.pone.0019070

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. (The noncentral t interval on the standardized mean difference that is mapped to the NNT here.) doi:10.18637/jss.v020.i08

Kraemer, H. C., & Kupfer, D. J. (2006). Size of treatment effects and their importance to clinical research and practice. *Biological Psychiatry, 59*(11), 990–996. doi:10.1016/j.biopsych.2005.09.014

See Also

smd, ci_smd, cles

Other effect size estimates: cles(), cliff_delta(), correction_for_attenuation(), eta_squared(), eta_squared_generalized(), eta_squared_partial(), expected_partial_r(), expected_r(), expected_smd(), omega_squared(), omega_squared_partial(), probability_of_superiority_paired(), proportion_of_superiority(), responder_analysis(), smd_trimmed()

Examples

```
# 1. Point estimate only.
nnt_from_smd(smd = 0.5)

# 2. With a noncentral t CI from sample sizes:
nnt_from_smd(smd = 0.5, n_1 = 50, n_2 = 50, conf_level = 0.95)

# 3. With pre-computed CI on d:
nnt_from_smd(smd = 0.5, smd_lower = 0.20, smd_upper = 0.80)

# 4. Lower d below zero: upper NNT bound is a finite negative value (NNT to harm).
nnt_from_smd(smd = 0.3, smd_lower = -0.10, smd_upper = 0.70)
```

---

now *A Friendly Date / Time Stamp and a Simple Stopwatch*

Description

now() is a small utility for two related tasks: (1) printing the current date and time in a form that reads naturally in a console log or report ("September 21, 2026 (3:42 PM)"), and (2) measuring how long a piece of work takes by capturing the time before and after the work and subtracting the two stamps.

Usage

```
now(time = TRUE, tidy = FALSE)
```

Arguments

time	Logical; if TRUE (default) the hour, minute, and AM/PM are appended to the printed date. FALSE returns the date only.
tidy	Logical; if TRUE, the timestamp is returned as a data.frame with columns term and value (numeric components only: day, year, hour, minute) and the non-numeric components (month name, AM/PM) attached as attributes. Defaults to FALSE, which returns the printable dmar_now stamp. Use tidy = TRUE only when the components are needed individually for programmatic processing; the dmar_now default supports subtraction and prettily prints in any context.

Details

The function returns an object of class "dmar_now" that carries the underlying Sys.time value and prints in the human-readable form. Two dmar_now objects can be subtracted with the ordinary - operator; the result is a difftime object with units chosen automatically from the magnitude of the elapsed interval. The pattern is the R analog of a stopwatch: capture before, capture after, take the difference.

For analyses that should record *when* a long-running computation was performed (a bootstrap confidence interval, a Monte Carlo simulation, an ss_aipe_*_sensitivity run), inserting a now() call at the start and end of the work creates a self-documenting log of when the computation ran and how long it took. The print method's natural-language form is designed to be copy-pasted into a methods section or an analysis note without further formatting.

The name coincides with lubridate::now(), and the masking is deliberate: both functions return the current time as a POSIXct object, so a script written for either remains correct with the other attached; only the printed form differs.

Value

When tidy = FALSE (default), an object of class c("dmar_now", "POSIXct", "POSIXt") whose print method yields the natural-language form ("September 21, 2026 (3:42 PM)" or, with time = FALSE, "September 21, 2026"). The underlying numeric value is the Sys.time() stamp at the moment of the call, so the - operator gives the elapsed time between two captures as a difftime object.

When tidy = TRUE, a data.frame with columns term and value, where value is numeric (day, year, hour, minute) and the month name and AM/PM marker are attached as the "month" and "am_pm" attributes.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

Sys.time for the underlying timestamp; difftime for the elapsed-time class returned by the - operator.

Examples

```
# Print the current date and time.
now()

# Time how long a piece of work takes. The pattern is the same
# whether the work is a bootstrap, a simulation, or a numeric
# search: capture a stamp before, capture one after, subtract.
# The "-" operator returns the elapsed time as a difftime, with
# units chosen automatically. Here the work is a descriptive
# summary of three Holzinger and Swineford cognitive tests, small
# enough that the elapsed time is a fraction of a second.
start <- now()
d <- descriptives(holzinger_swineford[, c("t1_visual_perception",
```

```
                                          "t2_cubes", "t4_lozenges")])
end <- now()
end - start

# A deliberate wait shows the same pattern on a longer interval. The
# pause is a fifth of a second, long enough to register in the
# difference and short enough not to slow the help page.
start <- now()
Sys.sleep(0.2)
end <- now()
end - start

# Date only.
now(time = FALSE)

# Tidy data.frame form, when the components are needed
# individually for programmatic processing, for example when the
# stamp is embedded in a report's metadata block.
now(tidy = TRUE)
```

`obrien_test` *O'Brien's Test for Homogeneity of Variance*

Description

Tests the null hypothesis that two or more groups have equal population variances using O'Brien's (1981) procedure: each observation is transformed into a quantity whose expected value equals the group's variance, and a one-way analysis of variance is then run on those transformed values. The test is generally regarded as more robust to non-normality than Bartlett's test while retaining good power.

Usage

```
obrien_test(x, group = NULL, data = NULL, na_action = stats::na.omit)
```

Arguments

`x`	Either a numeric vector of observations (in which case `group` must also be supplied), or a one-sided formula of the form `y ~ group`, in which case `data` is consulted for the variables.
`group`	A grouping vector or factor of the same length as `x`; used only when `x` is a numeric vector.
`data`	An optional `data.frame` containing the variables named in the formula.
`na_action`	Function specifying how missing values are handled (default `na.omit`).

Details

Following O'Brien (1981) and the version given in Abdi (2007), each observation Y_{ij} (the jth observation in group i, with size n_i and sample variance s_i^2) is transformed to

$$r_{ij} = \frac{(n_i - 1.5)\, n_i\, (Y_{ij} - \bar{Y}_i)^2 - 0.5\, s_i^2\, (n_i - 1)}{(n_i - 1)(n_i - 2)}.$$

The mean of the r_{ij} within group i equals s_i^2, so a one-way ANOVA on the r_{ij} tests $H_0 : \sigma_1^2 = \cdots = \sigma_k^2$. Each group must have at least three observations for the transformation to be defined.

Value

A one-row `data.frame` with columns `statistic` (the F-value from the ANOVA on the transformed scores), `df_1`, `df_2`, `p_value`, `n_groups`, `n_total`, and `method`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Abdi, H. (2007). O'Brien's test for homogeneity of variance. In N. J. Salkind (Ed.), *Encyclopedia of measurement and statistics*. Sage.

O'Brien, R. G. (1981). A simple test for variance effects in experimental designs. *Psychological Bulletin, 89*(3), 570–574.

See Also

`bartlett.test`, `var.test`

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `correlations_test()`, `equivalence_r()`, `equivalence_smd()`, `factorial_anova()`, `manova_split_plot()`, `mauchly_test()`, `mixed_anova()`, `pairwise_within()`, `randomization_test()`, `randomization_test_paired()`, `regions_of_significance()`, `simple_effects_AB()`, `summary_t_test()`, `welch_t()`

Examples

```
# Hunter's (1964) "one-is-a-bun" peg-word memory experiment, as discussed
# by Abdi (2007). Sixty-four participants were assigned to a control group
# (no mnemonic instruction) or an experimental group (peg-word mnemonic).
# The score is the number of word pairs (out of 10) recalled. Abdi (2007,
# Table 6) reports F = 1.29 (df = 1, 62) for the O'Brien test of equal
# variances, p = .260 as computed here; the experimental group's apparent
# ceiling effect does not produce statistically detectable variance
# heterogeneity.
hunter_1964 <- data.frame(
  group = factor(
    c(rep("Control", 32), rep("Experimental", 32)),
    levels = c("Control", "Experimental")
  ),
```

```
  recall = c(
    # Control group (n = 32):
    5, 5, 5, 5, 5,
    6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
    7, 7, 7, 7, 7, 7, 7, 7, 7,
    8, 8, 8,
    9, 9,
    10, 10,
    # Experimental group (n = 32):
    6,
    7, 7,
    8, 8, 8, 8,
    9, 9, 9, 9, 9, 9, 9, 9, 9,
    10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10
  )
)
obrien_test(recall ~ group, data = hunter_1964)

# Comparison against Bartlett's test on the same data.
bartlett.test(recall ~ group, data = hunter_1964)

# Vector / grouping-variable interface, on DMAR's depression_bdi data.
# The wait list variance is about twice the SSRI variance, but with ten
# observations per group the test does not reject equal variances.
obrien_test(depression_bdi$bdi_post, depression_bdi$condition)
```

`omega_squared` *Omega Squared (Effect Size for ANOVA)*

Description

Computes the sample omega squared (ω^2), Hays' (1994) bias-corrected estimator of the proportion of variance in the dependent variable accounted for by a fixed effect. Accepts either the raw ANOVA summary (F, the effect and error degrees of freedom, and total N) or a fitted `aov` or `lm` object, in which case the function returns one row per effect (partial ω^2 in factorial designs).

Usage

```
omega_squared(
  object = NULL,
  F_value = NULL,
  df_effect = NULL,
  df_error = NULL,
  N = NULL
)
```

Arguments

object	Optional. A fitted aov or lm object. When supplied, the function loops over the non-Residuals rows of anova(object) and returns one row per effect.
F_value	Observed F-value from the fixed-effects ANOVA (ignored if object is supplied).
df_effect	Numerator degrees of freedom for the effect (ignored if object is supplied).
df_error	Error (residual) degrees of freedom (ignored if object is supplied).
N	Total sample size (ignored if object is supplied; nobs(object) is used instead).

Details

The confidence interval is provided by the separate ci_omega_squared, paralleling the existing smd/ci_smd and eta_squared/ci_eta_squared pairings.

Point estimate. The reported value is Hays' (1994) sample omega squared, which for a one-way design is

$$\hat{\omega}^2 = \frac{df_{\text{effect}}(F-1)}{df_{\text{effect}}(F-1)+N}.$$

For factorial designs the same formula applied per effect yields *partial* omega squared (Olejnik & Algina, 2003); negative values are truncated to zero. This is the same point-estimate convention used by ci_omega_squared, so the two functions agree on the point estimate row by row.

Hand-in-hand with ci_omega_squared(). Pair this function with ci_omega_squared when reporting effect sizes: omega_squared() returns the point estimate(s), and ci_omega_squared() returns the same point estimate plus its noncentrality-based confidence limits (Steiger, 2004). The columns shared by the two functions (effect, omega_squared, F_value, df_effect, df_error, N) are aligned so the outputs compose cleanly with merge() or a join.

Sums of squares in factorial designs. anova() on an aov/lm uses Type I (sequential) sums of squares. For balanced designs all three types agree; for unbalanced designs they differ. If Type II or III F-values are required, compute them with e.g.\ car::Anova(object, type = 3) and pass the relevant F and degrees of freedom into the raw-argument interface.

Value

A data.frame with one row per effect. The columns are effect, omega_squared (point estimate), F_value, df_effect, df_error, and N. When the raw-argument interface is used, effect is "overall".

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Hays, W. L. (1994). *Statistics* (5th ed.). Fort Worth, TX: Harcourt Brace College Publishers.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*, 137–152. doi:10.1037/a0028086

Keppel, G. (1991). *Design and analysis: A researcher's handbook* (3rd ed.). Prentice Hall.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on η^2, Chapter 7 on factorial designs, and Chapter 11 on generalized η^2 for within-subjects designs.)

Olejnik, S., & Algina, J. (2003). Generalized eta and omega squared statistics: Measures of effect size for some common research designs. *Psychological Methods, 8*(4), 434–447. doi:10.1037/1082989X.8.4.434

Steiger, J. H. (2004). Beyond the *F* test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

ci_omega_squared, eta_squared, ci_eta_squared, ci_pvaf

Other effect size estimates: cles(), cliff_delta(), correction_for_attenuation(), eta_squared(), eta_squared_generalized(), eta_squared_partial(), expected_partial_r(), expected_r(), expected_smd(), nnt_from_smd(), omega_squared_partial(), probability_of_superiority_paired(), proportion_of_superiority(), responder_analysis(), smd_trimmed()

Examples

```
# 1. Raw-argument interface. Bargman's (1970) 5-group one-way ANOVA,
#        also used in Venables (1975), Fleishman (1980), and Steiger (2004):
#        11 subjects per group, observed F = 11.221.
omega_squared(F_value = 11.221, df_effect = 4, df_error = 50, N = 55)

# 2. One way ANOVA from a fitted model (depression_bdi: three
#        treatment arms, 10 per arm, N = 30).
fit_one <- aov(bdi_post ~ condition, data = depression_bdi)
omega_squared(fit_one)

# 3. Two-factor ANOVA: partial omega squared per effect for the
#        manipulated expectancy treatment and the measured grade
#        classification (pygmalion data, unequal cell sizes,
#        N = 310). The treatment by grade interaction is weak here
#        (F = 1.19), so the additive model is used.
fit_additive <- aov(iq_8 ~ treatment + factor(grade), data = pygmalion)
omega_squared(fit_additive)

# 4. omega_squared() and ci_omega_squared() compose: the point
#        estimates agree row-by-row.
pt  <- omega_squared(fit_additive)
ci  <- ci_omega_squared(fit_additive)
merge(pt, ci, by = c("effect", "omega_squared",
                     "F_value", "df_effect", "df_error", "N"))
```

`omega_squared_partial` *Partial Omega Squared (Effect Size for ANOVA)*

Description

Computes the sample *partial* omega squared (ω_p^2), Hays' (1994) bias-corrected estimator of the proportion of population variance in the dependent variable accounted for by a fixed effect after the variance attributable to the other effects in the model has been removed:

$$\hat{\omega}_p^2 = \frac{SS_{\text{effect}} - df_{\text{effect}} \cdot MS_{\text{error}}}{SS_{\text{effect}} + (N - df_{\text{effect}}) \cdot MS_{\text{error}}} = \frac{df_{\text{effect}}(F-1)}{df_{\text{effect}}(F-1) + N}.$$

Accepts either the raw ANOVA summary (F, effect df, error df, total N) or a fitted `aov`/`lm`/`aovlist` object, in which case the function returns one row per effect (with stratum identification for within-subjects fits).

Usage

```
omega_squared_partial(
  object = NULL,
  F_value = NULL,
  df_effect = NULL,
  df_error = NULL,
  N = NULL
)
```

Arguments

`object`	Optional. A fitted model object of class `aov`, `lm`, or `aovlist` (multi-stratum aov fit, e.g.\ `aov(y ~ A + Error(subject/A), data = d)`). When supplied, the function loops over the non-`Residuals` rows and returns one row per effect.
`F_value`	Observed F-value from the fixed-effects ANOVA (ignored if `object` is supplied).
`df_effect`	Numerator degrees of freedom for the effect (ignored if `object` is supplied).
`df_error`	Error (residual) degrees of freedom (ignored if `object` is supplied).
`N`	Total sample size (ignored if `object` is supplied; `nobs(object)` is used instead).

Details

This function is the explicitly-named counterpart of `omega_squared`. The two share the same point-estimate formula: in a one-way ANOVA they coincide with the total ω^2; in a factorial ANOVA both return the per-effect *partial* value computed against the model's residual mean square. `omega_squared_partial` is provided so that user code that explicitly intends partial ω^2 carries that meaning in its name, parallel to the `eta_squared` / `eta_squared_partial` pair.

Why partial omega squared and not total. In a one-way ANOVA, partial ω^2 reduces to total ω^2; in a factorial ANOVA the two diverge. Total ω^2 for an effect divides its variance contribution by the

total population variance of Y, so adding orthogonal factors to a study mechanically shrinks each effect's total ω^2. Partial ω^2 divides instead by the variance that is left after the other effects in the model have been partialled out, so a given fixed effect's partial ω^2 is (approximately) invariant to whether additional orthogonal factors are present (Olejnik & Algina, 2003; Maxwell, Delaney, & Kelley, 2027, Sections 7.4.4 and 8.4). For that reason, partial ω^2 is the chapter's preferred effect size index when off-factors are "extrinsic" (i.e., would not vary in a hypothetical full replication of the population setup).

Bias correction vs.\ partial eta squared. $\hat{\eta}_p^2$, the sample partial eta squared, is the proportion of *sample* variance accounted for and is upward-biased as an estimator of the population η_p^2. $\hat{\omega}_p^2$ subtracts $df_{\text{effect}} \cdot MS_{\text{error}}$ from the effect's sum of squares and rescales, yielding an estimator of the *population* variance proportion with substantially smaller bias (Hays, 1994; Olejnik & Algina, 2000; Kelley, 2007). Truncation at zero is conventional when the unbiased estimator goes negative because $\omega^2 \geq 0$ by definition.

Hand-in-hand with `ci_omega_squared()`. Pair this function with `ci_omega_squared` when reporting effect sizes: `omega_squared_partial()` returns the point estimate(s) and `ci_omega_squared()` returns the same point estimate plus its noncentral *F* confidence limits (Steiger, 2004; Kelley, 2007). The columns shared by the two functions are aligned so the outputs compose cleanly with `merge()` or a join.

Sums of squares in unbalanced factorial designs. `anova()` on an `aov`/`lm` uses Type I (sequential) sums of squares. For balanced designs all three SS types agree; for unbalanced designs they differ. If Type II or III *F*-values are required, compute them with e.g.\ `car::Anova(object, type = 3)` and pass the relevant *F* and degrees of freedom into the raw-argument interface.

Value

A `data.frame` with one row per effect. The columns are `effect`, `omega_squared_partial` (point estimate), `F_value`, `df_effect`, `df_error`, and `N`. When the raw-argument interface is used, `effect` is `"overall"`. Negative point estimates (which occur whenever $F < 1$) are truncated to zero, matching the convention used by `omega_squared` and `ci_omega_squared`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1973). Eta-squared and partial eta-squared in fixed factor ANOVA designs. *Educational and Psychological Measurement, 33*(1), 107–112.

Hays, W. L. (1994). *Statistics* (5th ed.). Fort Worth, TX: Harcourt Brace College Publishers.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*, 137–152. doi:10.1037/a0028086

Keppel, G., & Wickens, T. D. (2004). *Design and analysis: A researcher's handbook* (4th ed.). Pearson Prentice Hall.

Keren, G., & Lewis, C. (1979). Partial omega squared for ANOVA designs. *Educational and Psychological Measurement, 39*(1), 119–128.

Maxwell, S. E., Camp, C. J., & Arvey, R. D. (1981). Measures of strength of association: A comparative examination. *Journal of Applied Psychology, 66*(5), 525–534.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on η^2, Chapter 7 on factorial designs, and Chapter 11 on generalized η^2 for within-subjects designs.)

Olejnik, S., & Algina, J. (2000). Measures of effect size for comparative studies: Applications, interpretations, and limitations. *Contemporary Educational Psychology, 25*(3), 241–286. doi:10.1006/ceps.2000.1040

Olejnik, S., & Algina, J. (2003). Generalized eta and omega squared statistics: Measures of effect size for some common research designs. *Psychological Methods, 8*(4), 434–447. doi:10.1037/1082989X.8.4.434

Steiger, J. H. (2004). Beyond the *F* test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

omega_squared, ci_omega_squared, eta_squared_partial, ci_eta_squared_partial

Other effect size estimates: cles(), cliff_delta(), correction_for_attenuation(), eta_squared(), eta_squared_generalized(), eta_squared_partial(), expected_partial_r(), expected_r(), expected_smd(), nnt_from_smd(), omega_squared(), probability_of_superiority_paired(), proportion_of_superiority(), responder_analysis(), smd_trimmed()

Examples

```
# 1. Raw-argument interface (Bargman 1970 / Steiger 2004 example):
#        five groups of 11, observed F = 11.221.
omega_squared_partial(F_value = 11.221, df_effect = 4, df_error = 50, N = 55)

# 2. Two-factor ANOVA: partial omega squared per effect on the
#        pygmalion data (expectancy treatment x grade, unequal cell
#        sizes, N = 310). The treatment is manipulated while grade is
#        a measured classification, and the partial value for each
#        effect removes the variance the other accounts for. The
#        treatment by grade interaction is weak here (F = 1.19), so
#        the additive model is used.
fit_additive <- aov(iq_8 ~ treatment + factor(grade), data = pygmalion)
omega_squared_partial(fit_additive)

# 3. omega_squared_partial() and ci_omega_squared() agree on the
#        point estimate row-by-row.
pt  <- omega_squared_partial(fit_additive)
ci  <- ci_omega_squared(fit_additive)
pt$omega_squared_partial
ci$omega_squared

# 4. The named pair: omega_squared() and omega_squared_partial()
#        report identical numbers in this design; the only difference
#        is the name of the value column, which makes the user's
```

```
#        intent (partial) explicit.
omega_squared(fit_additive)$omega_squared
omega_squared_partial(fit_additive)$omega_squared_partial
```

`orthogonal_polynomial` *Orthogonal-Polynomial (Trend) Contrast Coefficients*

Description

Builds the set of orthogonal-polynomial contrasts (linear, quadratic, cubic, ...) for a factor whose a levels are quantitative, so that between-group variation can be decomposed into independent trend components. The columns are mutually orthogonal and each sums to zero (so each is a contrast on the group means). By default the coefficients are returned in *orthonormal* form, meaning each column also has unit length, $\sum_i c_i^2 = 1$; the alternative `type = "integer"` rescales every column to the small whole numbers used in the published orthogonal-polynomial tables, which are easier to read by eye and match hand computation. The per-trend sum of squared coefficients $\sum_i c_i^2$ is carried on the returned object and shown when it is printed.

Usage

```
orthogonal_polynomial(
  levels,
  scores = NULL,
  type = c("orthonormal", "integer"),
  degree = NULL
)
```

Arguments

`levels`	One of: a single integer giving the number of levels a (labels default to `"L1"`, `"L2"`, ..., and the levels are treated as equally spaced); a character or factor vector of level labels (equally spaced unless `scores` is given); or a numeric vector of the quantitative level values themselves, which are used both as the labels and as the spacing.
`scores`	Optional numeric vector of length a giving the quantitative position of each level. Supply this when the labels are non-numeric but the spacing is unequal (e.g., doses 0, 1, 2, and 4). Defaults to equally spaced positions `1:a`.
`type`	Either `"orthonormal"` (default) for unit-length columns ($\sum_i c_i^2 = 1$, identical to `contr.poly`) or `"integer"` for the minimal whole-number coefficients of the classic trend-coefficient table. `"integer"` requires equally spaced `scores`.
`degree`	Highest-order trend to return, an integer between 1 and $a - 1$. Defaults to $a - 1$ (the full set the design can support).

Details

What a trend contrast is. When the levels of a factor are quantitative and ordered (minutes of study, dose, day), the omnibus between-group variation can be partitioned into a linear trend (does the mean rise or fall steadily?), a quadratic trend (is there curvature?), a cubic trend (an S-shape?), and so on, up to order $a - 1$. Each trend is a single 1-*df* contrast on the group means, $\hat{\psi} = \sum_i c_i \bar{Y}_i$, and because the contrasts are mutually orthogonal their sums of squares add up to the omnibus between-group sum of squares exactly.

Orthonormal versus integer scaling. The two `type` values return the *same* trends (same directions, same hypotheses, identical *F*, *t*, and *p* for a given trend); they differ only in how each column is scaled.

- `"orthonormal"` normalizes each column to unit length, $\sum_i c_i^2 = 1$. The trend portion of the design is then an orthonormal basis, the sum of squares for a trend is simply $\hat{\psi}^2$, and in a fitted `lm` the `.linear` / `.quadratic` coefficients are the trend estimates on a common scale. This is the form `lm` and `aov` use internally and is defined for any spacing, equal or unequal.
- `"integer"` rescales each column to the smallest whole numbers with the same ratios, reproducing the published orthogonal-polynomial coefficient table (e.g., for $a = 4$ the linear contrast is $-3, -1, 1, 3$). These are easy to read and to compute with by hand, but the columns no longer share a common length: $\sum_i c_i^2$ varies by trend (for $a = 4$: 20, 4, and 20), so the sum of squares for a trend is $n\,\hat{\psi}^2 / \sum_i c_i^2$. Integer coefficients exist only for equally spaced levels.

The reported $\sum_i c_i^2$ (shown on printing and stored in `attr(*, "sum_sq")`) is exactly the divisor in that sum-of-squares formula; for the orthonormal form it is 1 for every trend.

Unequal spacing. With unequally spaced `scores` the orthonormal polynomials are still uniquely defined and are returned by the `"orthonormal"` type. Whole-number coefficients generally do not exist in that case, so `type = "integer"` is an error.

Relation to base R. For equally spaced levels the orthonormal output is identical to `contr.poly(a)`; `orthogonal_polynomial` adds the integer table form, an explicit `scores` argument for unequal spacing, meaningful trend names, and the $\sum_i c_i^2$ report.

Value

A numeric $a \times$ degree matrix with row names = the level labels and column names = the trend names (`"linear"`, `"quadratic"`, `"cubic"`, `"quartic"`, ...). The matrix can be assigned directly to `contrasts(factor)` or passed to `contrast_test`, `is_orthogonal_set`, or `ci_c`. The per-column sum of squared coefficients is stored as `attr(*, "sum_sq")` (a named numeric vector), and the level spacing and `type` are stored as `attr(*, "scores")` and `attr(*, "type")`. The object carries class `"orthogonal_polynomial"` so that printing shows the coefficients alongside $\sum_i c_i^2$. When printed, the object is shown in the textbook Table A.10 orientation (trends in rows, levels in columns) with the $\sum_i c_i^2$ values as a final column; the stored matrix is the transpose of that display (levels in rows) so it can be assigned directly to `contrasts()`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Fisher, R. A., & Yates, F. (1953). *Statistical tables for biological, agricultural and medical research* (4th ed.). Oliver and Boyd. (Origin of the tabulated integer coefficients.)

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 6 on trend analysis; Appendix Table A.10 reports these coefficients.)

See Also

contr.poly, effects_coding, helmert_coding, is_orthogonal_set, contrast_test, ci_c

Other design utilities: design_consequences(), design_effect(), effects_coding(), helmert_coding(), is_orthogonal_set()

Examples

```
# Trend (orthogonal-polynomial) analysis decomposes the omnibus effect of a
# quantitative factor (dose, time, trial block, stimulus intensity) into
# independent linear, quadratic, cubic, ... components, as in the trend
# analysis of Maxwell, Delaney, and Kelley (2027, Chapter 6).

# 1. The a = 4 coefficient table in the integer form of the textbook
#    appendix (Table A.10). Printing puts the trends in rows and appends a
#    final sum-of-squared-coefficients (sum c^2) column: linear -3 -1 1 3,
#    and per-trend sum c^2 of 20, 4, 20.
orthogonal_polynomial(4, type = "integer")

# 2. The default orthonormal form (identical to stats::contr.poly(4)); every
#    trend has sum c^2 = 1, the scale lm() and aov() use internally.
orthogonal_polynomial(4)

# 3. A worked trend analysis in the style of MDK Chapter 6. An outcome is
#    measured at a = 5 equally spaced levels of a quantitative factor (say
#    stimulus intensity 1..5), n = 8 per level, from a population with a
#    strong linear and a mild quadratic trend. Assigning the trend contrasts
#    to the factor makes lm() report each trend test directly.
set.seed(113)
intensity <- factor(rep(1:5, each = 8))
contrasts(intensity) <- orthogonal_polynomial(levels(intensity))
y <- c(10, 9, 7, 6, 6)[as.integer(intensity)] +
       rnorm(length(intensity), sd = 1.2)
round(coef(summary(lm(y ~ intensity))), 3)

# 4. The same trends through DMAR's contrast_test(), which reports each
#    trend's estimate, standard error, t, p, and confidence interval from a
#    fitted one-way model. The integer columns are the contrast weights.
op <- orthogonal_polynomial(5, type = "integer")
contrast_test(aov(y ~ intensity),
              contrasts = list(linear    = op[, "linear"],
                               quadratic = op[, "quadratic"],
                               cubic     = op[, "cubic"]))
```

```
# 5. Unequally spaced doses (0, 1, 2, 4 mg): integer coefficients no longer
#    exist, but the orthonormal trends remain uniquely defined.
orthogonal_polynomial(c("0 mg", "1 mg", "2 mg", "4 mg"),
                      scores = c(0, 1, 2, 4))

# 6. Confirm a returned set is mutually orthogonal.
is_orthogonal_set(orthogonal_polynomial(5, type = "integer"))
```

`pairwise_within` *Paired Pairwise Comparisons With Multiple-Comparison Adjustment*

Description

Computes all pairwise paired-*t* comparisons among the levels of a within-subjects factor and returns the mean difference, paired SD, paired *t*-statistic, degrees of freedom, raw and adjusted *p*-values, and a confidence interval on the mean difference, all in tidy long form. *p*-values are adjusted across comparisons by the user-specified method (Bonferroni, Holm, Hochberg, Hommel, BH, BY, or none).

Usage

```
pairwise_within(
  data,
  subject = NULL,
  condition = NULL,
  outcome = NULL,
  adjust = c("holm", "bonferroni", "hochberg", "hommel", "BH", "BY", "none"),
  conf_level = 0.95,
  bonferroni_ci = FALSE
)
```

Arguments

`data`	Either an $n \times k$ wide numeric matrix / data.frame (one row per subject, one column per condition), or a long-format `data.frame` together with `subject`, `condition`, and `outcome` column names.
`subject`	Long-format only: character name of the subject-id column.
`condition`	Long-format only: character name of the within- subjects factor column.
`outcome`	Long-format only: character name of the response column.
`adjust`	Multiple-comparison adjustment method. One of `"holm"` (default), `"bonferroni"`, `"hochberg"`, `"hommel"`, `"BH"`, `"BY"`, or `"none"`. Passed to `stats::p.adjust()`.
`conf_level`	Family-wise confidence level for the per-pair CIs. Default `0.95`. CIs are computed at the per-pair nominal level ($1 - \alpha/m$ under Bonferroni; `conf_level` otherwise).
`bonferroni_ci`	Logical. If `TRUE`, the CIs use a per-pair confidence level of $1-(1-\text{conf\_level})/m$ to give simultaneous `conf_level` coverage across the m comparisons (Bonferroni-corrected CIs). Default `FALSE`.

Details

Why a paired pairwise. `stats::pairwise.t.test()` returns a square matrix of p-values, which doesn't compose with the rest of the **DMAR** pipeline. This function returns one row per comparison, matching the `data.frame(term, value)` style used elsewhere.

CI scale. CIs are on the mean-difference scale (unstandardized). When `bonferroni_ci = TRUE`, the per-pair confidence level is $1-(1-\text{conf\_level})/m$, giving Bonferroni-style simultaneous coverage. The CI is built from the paired-t distribution with $n-1$ degrees of freedom.

Value

A `data.frame` with one row per pair. Columns: `contrast` (the labeled difference, e.g. `"B - A"`), `mean_difference`, `sd_difference`, `t_statistic`, `df`, `p_value` (raw), `p_adjusted`, `lower_limit`, `upper_limit`, `n_pairs`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 11.)

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics, 6*(2), 65–70.

See Also

`pairwise.t.test`, `anova_within`

Other within-subjects analysis: `anova_within()`, `anova_within_two_way()`, `epsilon_corrections()`, `mauchly_test()`, `plot_trajectories_fitted()`

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `correlations_test()`, `equivalence_r()`, `equivalence_smd()`, `factorial_anova()`, `manova_split_plot()`, `mauchly_test()`, `mixed_anova()`, `obrien_test()`, `randomization_test()`, `randomization_test_paired()`, `regions_of_significance`
`simple_effects_AB()`, `summary_t_test()`, `welch_t()`

Examples

```
# 1. Wide-format input: 4 timepoints x 10 subjects.
set.seed(113)
n <- 10; k <- 4
Y <- matrix(rnorm(n * k, 0, 1), n, k) +
     matrix(rep(seq(0, 0.9, length.out = k), n), n, k, byrow = TRUE) +
     rnorm(n, 0, 1.5)
colnames(Y) <- paste0("T", 1:k)
pairwise_within(Y)

# 2. Long-format input:
long <- data.frame(
```

```
  subject = factor(rep(1:n, times = k)),
  time    = factor(rep(paste0("T", 1:k), each = n)),
  y       = as.vector(Y)
)
pairwise_within(long, subject = "subject", condition = "time", outcome = "y")
```

---

plot_cfa_k *Plot the Estimates of a Multiple-Factor CFA*

Description

Displays the item-level estimates of a cfa_k fit, one panel per factor, with each estimate's confidence interval. The display is built to make the equality questions behind the classical measurement structures visible: a dashed vertical line marks, per factor, either the common (equated) estimate when the plotted parameter was constrained equal, or the mean of the free estimates as an informal anchor for the question "could these plausibly be one value?". Confidence intervals that all cover the anchor are what equal loadings (or equal error variances, or equal intercepts) would look like; an interval far from it shows which item resists the constraint, and the likelihood ratio test of the two nested cfa_k() fits is the formal companion (see the examples in cfa_k).

Usage

```
plot_cfa_k(
  x,
  what = c("loadings", "errors", "intercepts"),
  show_equal_reference = TRUE,
  xlab = NULL,
  title = NULL,
  palette = "okabe_ito"
)
```

Arguments

x	A dmar_cfa_k object from cfa_k with the default output = "verbose".
what	Which parameter to display: "loadings" (default, the lambda terms), "errors" (the psi terms), or "intercepts" (the nu terms; requires a fit with the mean structure).
show_equal_reference	Logical. If TRUE (default), draw the dashed per-factor reference line described above. When the parameter was constrained equal the line is the common estimate and is always drawn.
xlab	Label for the horizontal axis. Defaults to a description of the plotted parameter.
title	Optional plot title.
palette	Character string naming the color palette. Defaults to "okabe_ito", base R's colorblind-safe Okabe-Ito palette; "tableau" is also available.

Value

A ggplot2 object.

Note

Requires **ggplot2** (listed in Suggests).

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

cfa_k for the fit; plot_ci for the general forest-style confidence interval display.

Other plotting: plot_R2(), plot_ci(), plot_equivalence(), plot_forest(), plot_irt_information(), plot_mediation_mbco(), plot_randomization_test(), plot_regions_of_significance(), plot_smd(), plot_trajectories(), plot_trajectories_fitted(), power_equivalence_md_plot()

Examples

```
data(holzinger_swineford)
hs_factors <- list(
  verbal = c("t6_paragraph_comprehension",
             "t7_sentence", "t9_word_meaning"),
  deduction = c("t20_deduction", "t22_problem_reasoning",
             "t23_series_completion"))
res <- cfa_k(holzinger_swineford, hs_factors)

# Are equal loadings plausible? Compare each interval with the anchor.
plot_cfa_k(res)

# The same question for the error variances, the additional constraint
# that separates essentially parallel from essentially tau-equivalent.
plot_cfa_k(res, what = "errors")

# After imposing the constraint, every item in a factor sits at the
# common estimate and the dashed line is that estimate rather than
# the mean of the free ones.
res_equal <- cfa_k(holzinger_swineford, hs_factors,
                   equal_loading = TRUE)
plot_cfa_k(res_equal)
```

---

plot_ci *Forest-Plot-Style Confidence Interval Display*

Description

Creates a clean visualization of one or more effect size estimates with their confidence intervals.

Usage

```
plot_ci(
  ci = NULL,
  estimate = NULL,
  lower = NULL,
  upper = NULL,
  names = NULL,
  n = NULL,
  conf_level = 0.95,
  show_n = TRUE,
  reference_line = NULL,
  xlab = "Effect Size",
  title = NULL,
  palette = "okabe_ito"
)
```

Arguments

ci	A data.frame from a DMAR ci_* function. When supplied, the function auto-detects the format and extracts the point estimate(s), lower limit(s), and upper limit(s). Explicit estimate, lower, and upper arguments override values parsed from ci.
estimate	Numeric vector of point estimates.
lower	Numeric vector of lower confidence limits.
upper	Numeric vector of upper confidence limits.
names	Optional character vector of labels for each effect.
n	Optional numeric vector (or scalar) of sample sizes. Recycled to match the number of effects.
conf_level	Confidence level; used only for the axis label (default 0.95).
show_n	Logical. If TRUE (the default), the sample size is annotated above each estimate, with the estimate and its interval printed below.
reference_line	Optional numeric value at which to draw a vertical reference line (e.g., 0 for mean differences, 1 for ratios).
xlab	Label for the horizontal (effect size) axis. Defaults to "Effect Size".
title	Optional plot title.
palette	Character string naming the color palette. The point estimates and interval bars are drawn in the palette's primary color. Defaults to "okabe_ito", base R's colorblind-safe Okabe-Ito palette; "tableau" is also available.

Details

The function accepts either (a) a data.frame produced by an DMAR ci_* function (e.g., ci_smd, ci_R2, ci_omega_squared), or (b) explicit numeric vectors for the estimate(s), lower bound(s), and upper bound(s).

The function recognizes three DMAR output formats:

Long term/value with estimate row Output from ci_smd, which includes a row for the point estimate (e.g., term = "smd") in addition to "lower_limit" and "upper_limit".

Long term/value without estimate Output from ci_R or ci_R2, which contains only "lower_limit" and "upper_limit". Supply the point estimate via the estimate argument.

Wide per-effect format Output from ci_omega_squared, which has one row per effect with columns for the point estimate, lower_limit, upper_limit, and N.

Value

A ggplot2 object.

Note

Requires **ggplot2** (listed in Suggests).

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

ci_smd, ci_R, ci_R2, ci_omega_squared, plot_smd, plot_R2

Other plotting: plot_R2(), plot_cfa_k(), plot_equivalence(), plot_forest(), plot_irt_information(), plot_mediation_mbco(), plot_randomization_test(), plot_regions_of_significance(), plot_smd(), plot_trajectories(), plot_trajectories_fitted(), power_equivalence_md_plot()

Examples

```
# From explicit values.
plot_ci(estimate = 0.45, lower = 0.15, upper = 0.75,
        names = "Cohen's d", n = 60, reference_line = 0)

# From ci_smd() output.
ci_result <- ci_smd(smd = 0.5, n_1 = 50, n_2 = 50)
plot_ci(ci_result, n = 100, reference_line = 0)

# Multiple effects from ci_omega_squared(): the expectancy treatment
# and the grade classification in the pygmalion data.
pyg <- pygmalion
pyg$grade <- factor(pyg$grade)
fit <- aov(iq_8 ~ treatment + grade, data = pyg)
omega_result <- ci_omega_squared(fit)
plot_ci(omega_result, reference_line = 0,
        xlab = expression(omega^2))
```

`plot_equivalence` *Plot Contrasts Against an Equivalence Region*

Description

Draws a forest-style plot of one or more contrast estimates with their 100(1 - 2α)% confidence intervals against the equivalence region $(-\delta_L, \delta_U)$ and the noninferiority bound $-\delta_L$, colored by the five-way verdict of `equivalence_c`: an interval entirely inside the region is equivalent; entirely above δ_U, superior; entirely below $-\delta_L$, inferior; a lower limit above $-\delta_L$ with an upper limit past δ_U, noninferior only; and an interval straddling a bound, inconclusive. The geometry *is* the decision rule, which is what makes the plot the natural report of an equivalence analysis.

Usage

```
plot_equivalence(
  x = NULL,
  estimate = NULL,
  lower = NULL,
  upper = NULL,
  names = NULL,
  delta_lower = NULL,
  delta_upper = NULL,
  xlab = "Contrast",
  title = NULL,
  palette = "okabe_ito"
)
```

Arguments

`x`	Either a single result from `equivalence_c` or a list of them (a named list supplies the row labels). Alternatively, supply `estimate`, `lower`, and `upper` directly.
`estimate, lower, upper`	Numeric vectors of contrast estimates and their confidence limits, used when `x` is not supplied.
`names`	Optional character vector of row labels.
`delta_lower, delta_upper`	Equivalence bounds, as positive magnitudes (the region drawn is $(-\delta_L, +\delta_U)$). Taken from `x` when it carries `equivalence_c` results; required otherwise. If only `delta_upper` is supplied, the bounds are symmetric.
`xlab`	The horizontal axis label. Default `"Contrast"`.
`title`	Optional plot title.
`palette`	Character string naming the color palette for the verdict colors. Defaults to `"okabe_ito"`, base R's colorblind-safe Okabe-Ito palette; `"tableau"` is also available.

Details

The shaded band is the equivalence region and the dashed vertical lines are its bounds; the solid line at zero marks exact equality, which is the null value of ordinary significance testing and is deliberately *not* a decision boundary here. Verdicts are recomputed from the supplied limits and bounds, so the plot cannot disagree with `equivalence_c`.

Value

A `ggplot` object. Requires **ggplot2** to be installed.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Chattopadhyay, B., Bandyopadhyay, T., Kelley, K., & Padalunkal, J. J. (2025). A sequential approach for noninferiority or equivalence of a linear contrast under cost constraints. *Psychological Methods, 30*(2), 425–439. doi:10.1037/met0000570

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics, 15*(6), 657–680.

See Also

`equivalence_c`, `plot_ci`

Other equivalence testing: `equivalence_c()`, `equivalence_r()`, `equivalence_smd()`, `power_density_equivalence_md` `power_equivalence_c()`, `power_equivalence_md()`, `power_equivalence_md_plot()`, `ss_power_equivalence_c()`

Other plotting: `plot_R2()`, `plot_cfa_k()`, `plot_ci()`, `plot_forest()`, `plot_irt_information()`, `plot_mediation_mbco()`, `plot_randomization_test()`, `plot_regions_of_significance()`, `plot_smd()`, `plot_trajectories()`, `plot_trajectories_fitted()`, `power_equivalence_md_plot()`

Examples

```
# Five constructed intervals, one per verdict, against bounds of 5
# (A equivalent, B noninferior only, C superior, D inconclusive,
#  E inferior):
plot_equivalence(estimate = c(-1.0, 3.5, 7.0, -1.5, -8.0),
                 lower    = c(-3.2, -1.4, 5.5, -6.6, -10.5),
                 upper    = c( 1.2,  8.4, 8.5,  3.6,  -5.5),
                 names    = c("A", "B", "C", "D", "E"),
                 delta_upper = 5)

# From equivalence_c() results; a named list supplies the labels.
res <- list(
  "Focal vs. reference" = equivalence_c(psi_hat = -5.28, se = 2.49,
                                        df_error = 399, delta_upper = 5),
  "Within pipeline"     = equivalence_c(psi_hat = -0.53, se = 2.66,
                                        df_error = 399, delta_upper = 5)
```

```
)
plot_equivalence(res)
```

`plot_forest` *Forest Plot of Study Effect Sizes With the Pooled Estimate*

Description

Draws the meta-analyst's central picture: every study's effect size with its confidence interval, the random effects pooled estimate beneath them, and, by default, the prediction interval showing where the effect of a *new* study is expected to land. Point sizes are proportional to precision (inverse variance), so the eye weighs the studies the way the model does. Requires **ggplot2**.

Usage

```
plot_forest(
  yi,
  vi,
  labels = NULL,
  method = c("reml", "pm", "dl", "fe"),
  hartung_knapp = TRUE,
  conf_level = 0.95,
  show_prediction = TRUE,
  xlab = "Effect size",
  title = NULL,
  palette = "okabe_ito",
  colors = NULL
)
```

Arguments

`yi`	Numeric vector of study effect sizes.
`vi`	Sampling variances of `yi`.
`labels`	Optional study labels, one per study; defaults to `"Study 1"`, `"Study 2"`, and so on, in the supplied order.
`method, hartung_knapp`	Passed to `meta_es` for the pooled row (`"reml"` and `TRUE` by default).
`conf_level`	Confidence level for the per-study and pooled intervals. Defaults to 0.95.
`show_prediction`	Logical: draw the prediction interval band on the pooled row? Default `TRUE` (ignored for `method = "fe"`, which has none).
`xlab`	Label for the effect size axis. Defaults to `"Effect size"`.
`title`	Optional plot title.

palette	Palette name. Defaults to "okabe_ito", base R's colorblind-safe Okabe-Ito palette; "tableau" is also available.
colors	Optional length-2 vector overriding the palette: the study color and the pooled-estimate color.

Value

A **ggplot** object; print it, or add further layers.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

meta_es, meta_smd, and meta_r for the numbers behind the picture; teacher_expectancy for the example data.

Other meta-analysis: combine_p(), meta_contrast(), meta_es(), meta_r(), meta_smd()

Other plotting: plot_R2(), plot_cfa_k(), plot_ci(), plot_equivalence(), plot_irt_information(), plot_mediation_mbco(), plot_randomization_test(), plot_regions_of_significance(), plot_smd(), plot_trajectories(), plot_trajectories_fitted(), power_equivalence_md_plot()

Examples

```
# Twelve simulated studies whose true effects vary from study to study
# (between-study standard deviation 0.35), so the prediction interval
# for the effect of a new study is visibly wider than the confidence
# interval for the mean effect.
set.seed(113)
k <- 12
n <- sample(20:100, k)                    # per-group sample sizes
theta <- rnorm(k, mean = 0.4, sd = 0.35)  # true study effects
d <- rnorm(k, mean = theta, sd = sqrt(2 / n))
v <- 2 / n + d^2 / (4 * n)
plot_forest(d, v, xlab = "Standardized mean difference (d)")

# The teacher expectancy literature (Raudenbush, 1984): most studies
# cluster near zero, the estimated between-study variance is zero, and
# the prediction interval nearly coincides with the confidence interval.
data(teacher_expectancy)
d <- teacher_expectancy$d
n_e <- teacher_expectancy$n_experimental
n_c <- teacher_expectancy$n_control
v <- (n_e + n_c) / (n_e * n_c) + d^2 / (2 * (n_e + n_c))
plot_forest(d, v, labels = teacher_expectancy$author,
            xlab = "Standardized mean difference (d)")
```

`plot_irt_information` *Plot an Item Response Theory Information Curve*

Description

Draws the information function computed by `irt_information`: either the test information curve, with the standard error of the latent trait estimate on a secondary axis, or one curve per item. The test view answers "where on the latent continuum does this scale measure precisely?", and because the standard error is $1/\sqrt{I(\theta)}$ the same picture shows the precision directly. The item view decomposes that curve, since information is additive across items, and so shows which items cover which part of the continuum.

Usage

```
plot_irt_information(
  x,
  what = c("test", "item"),
  show_se = TRUE,
  show_peak = TRUE,
  palette = "okabe_ito",
  title = NULL,
  xlab = NULL,
  ylab = NULL
)
```

Arguments

`x`	The result of `irt_information`.
`what`	Which curves to draw: `"test"` (default) for the test information function, or `"item"` for one curve per item.
`show_se`	Logical. When `TRUE` (the default) and `what = "test"`, the standard error of the latent trait estimate is drawn as a dashed curve against a secondary axis. The layer is omitted when the standard error is not finite and varying over the grid (for example when test information is zero somewhere).
`show_peak`	Logical. When `TRUE` (the default) and `what = "test"`, a vertical dotted line marks the value of `theta` at which test information peaks on the supplied grid.
`palette`	Character string naming the color palette. Defaults to `"okabe_ito"`, base R's colorblind-safe Okabe-Ito palette; `"tableau"` is also available.
`title`	Optional plot title.
`xlab`	Label for the horizontal axis. Defaults to a description of the latent trait metric.
`ylab`	Label for the vertical axis. Defaults to a description of the information plotted.

Details

The secondary axis is a linear rescaling of the primary axis, so the dashed standard error curve shares the panel with the information curve without either being distorted relative to its own axis. The standard error is largest where information is smallest, which is why the two curves run in opposite directions.

Value

A `ggplot2` object.

Note

Requires **ggplot2** (listed in `Suggests`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Embretson, S. E., & Reise, S. P. (2000). *Item response theory for psychologists*. Lawrence Erlbaum.

Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph Supplement, 34*(4, Pt. 2), 1–97.

See Also

`irt_information`

Other plotting: `plot_R2()`, `plot_cfa_k()`, `plot_ci()`, `plot_equivalence()`, `plot_forest()`, `plot_mediation_mbco()`, `plot_randomization_test()`, `plot_regions_of_significance()`, `plot_smd()`, `plot_trajectories()`, `plot_trajectories_fitted()`, `power_equivalence_md_plot()`

Examples

```
info <- irt_information(
  a = c(mood_1 = 1.4, mood_2 = 0.9, mood_3 = 1.1),
  b = c(-1.5, -0.5, 0.5, 1.5, 0.0, 0.8),
  item = c(rep("mood_1", 4), "mood_2", "mood_3")
)

# Test information with the standard error on the secondary axis.
plot_irt_information(info)

# One curve per item.
plot_irt_information(info, what = "item")
```

plot_mediation_mbco *Plot Conditional Effects From a Moderated Mediation Analysis*

Description

Draws the conditional effects from a mediation_mbco analysis that declared a moderator: for each moderated pathway effect, the curve tracing how the effect changes over the moderator's range, with a pointwise confidence band, the probed values marked, a dashed reference line at zero, and a rug showing where the moderator was actually observed. The picture answers, at a glance, the questions the table answers row by row: how large is the effect at any given moderator value, where (if anywhere) does its interval exclude zero, and over what part of the moderator's range the data can support either statement.

Usage

```
plot_mediation_mbco(
  x,
  effects = NULL,
  conf_level = NULL,
  B = 10000,
  from = NULL,
  to = NULL,
  n_grid = 200,
  show_probe_values = TRUE,
  show_rug = TRUE,
  palette = c("okabe_ito", "tableau"),
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  seed = NULL
)
```

Arguments

x	A dmar_mediation_mbco object returned by mediation_mbco with a moderator. An object fit without a moderator has no conditional effects to draw, and the function says so.
effects	Character vector naming which moderated effects to draw, using the base effect names from the result table (e.g., "indirect_via_m", "total_effect"). Defaults to all moderated effects. Unmoderated effects are flat lines and are not drawn.
conf_level	Confidence level for the band. Defaults to the level used when the object was fit.
B	Number of Monte Carlo draws behind the band. Defaults to 10000.

`from`, `to` Range of moderator values to draw. Defaults to the observed range of the moderator. Values outside the observed range are extrapolation; the rug makes that visible.

`n_grid` Number of grid points along the moderator at which the curve and band are evaluated. Defaults to 200.

`show_probe_values` Logical. If `TRUE` (default), mark the probed moderator values (the `_at_` rows of the result table) as points on each curve.

`show_rug` Logical. If `TRUE` (default), draw a rug of the observed moderator values along the horizontal axis.

`palette` Character string naming the color palette. Defaults to `"okabe_ito"`, base R's colorblind-safe Okabe-Ito palette; `"tableau"` is also available.

`xlab`, `ylab`, `title` Optional axis labels and title. The defaults name the moderator on the horizontal axis and describe the vertical axis as the conditional effect of `x` on `y`.

`seed` Optional integer seed for the Monte Carlo band, used locally (the caller's random number generator state is restored on exit). Default `NULL` leaves the random number generator state alone.

Details

What is drawn, and where it comes from. A pathway effect in a model with interactions is a polynomial in the moderator: a straight line when the pathway is moderated in one place (its slope is the index of moderated mediation), a curve when it is moderated in more than one. `mediation_mbco` derives each polynomial symbolically and stores it with the result, so this function evaluates the same quantity the table probes, just everywhere in the moderator's range instead of at two or three values. The marked points are exactly the table's `_at_` rows.

The band is pointwise. At each grid value of the moderator, the band is a `conf_level` Monte Carlo confidence interval for the conditional effect at that one value: the path coefficients are drawn from their joint normal approximation (MacKinnon, Lockwood, & Williams, 2004), each draw's polynomial is evaluated along the grid, and the band connects the pointwise quantiles. Read vertically at a single moderator value of interest, it is an ordinary confidence interval. Read horizontally, the moderator values where the band crosses zero estimate the Johnson-Neyman boundaries (Johnson & Neyman, 1936; Preacher, Rucker, & Hayes, 2007), the values separating "interval excludes zero" from "interval includes zero". That horizontal reading scans many intervals at once, so the pointwise band understates the uncertainty of the boundary locations themselves; treat the crossing points as estimates, not as sharp cutoffs, and lean on the table's moderation and constancy tests for the formal question of whether the effect depends on the moderator at all.

The rug guards against extrapolation. The curve can be evaluated at any moderator value, but the data only inform it where the moderator was observed. The rug shows that support directly; a confident-looking band in a region with no rug beneath it is arithmetic, not evidence.

The band and the table may differ slightly. The band is always Monte Carlo, whichever `ci_method` the table used. At a probed value, a Monte Carlo band and a profile likelihood or Wald interval agree closely in large samples but are not the same construction; small discrepancies between the band and an `_at_` row's interval are expected, not a defect.

The plot is an ordinary **ggplot2** object, so any further customization (themes, additional layers, institutional color scales) can be added to the returned value with `+`.

Value

A ggplot2 object. Its data contains one row per effect and grid value with columns effect_label, w_value, estimate, band_lower, and band_upper, so the numbers behind the picture are recoverable from the object itself.

Note

Requires **ggplot2** (listed in Suggests).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Johnson, P. O., & Neyman, J. (1936). Tests of certain linear hypotheses and their application to some educational problems. *Statistical Research Memoirs, 1*, 57–93.

MacKinnon, D. P., Lockwood, C. M., & Williams, J. (2004). Confidence limits for the indirect effect: Distribution of the product and resampling methods. *Multivariate Behavioral Research, 39*(1), 99–128. doi:10.1207/s15327906mbr3901_4

Preacher, K. J., Rucker, D. D., & Hayes, A. F. (2007). Addressing moderated mediation hypotheses: Theory, methods, and prescriptions. *Multivariate Behavioral Research, 42*(1), 185–227. doi:10.1080/00273170701341316

Tofighi, D., & Kelley, K. (2020). Improved inference in mediation analysis: Introducing the model-based constrained optimization procedure. *Psychological Methods, 25*(4), 496–515. doi:10.1037/met0000259

See Also

mediation_mbco for the analysis this function displays; regions_of_significance for the analogous display for mixed-effects model interactions.

Other plotting: plot_R2(), plot_cfa_k(), plot_ci(), plot_equivalence(), plot_forest(), plot_irt_information(), plot_randomization_test(), plot_regions_of_significance(), plot_smd(), plot_trajectories(), plot_trajectories_fitted(), power_equivalence_md_plot()

Examples

```
# First-stage moderated mediation: the effect of x on m depends on
# w, so the indirect effect of x on y through m is a line in w. The
# simulated x reaches y only through m, and the model fit below
# carries no direct path, so the indirect effect is the whole effect
# of x and the picture has one curve.
set.seed(113)
n <- 300
x <- rnorm(n)
w <- rnorm(n)
m <- 0.5 * x + 0.3 * w + 0.4 * x * w + rnorm(n)
y <- 0.5 * m + 0.1 * w + rnorm(n)
d_mod <- data.frame(x = x, w = w, m = m, y = y)
```

```
# Fit with the moderator declared. Every reported row costs its own
# constrained null model fit in OpenMx, so the Wald interval and two
# probe values keep the fit quick; the default probe values are the
# moderator's mean and one standard deviation either side, and the
# curve and its band cover the whole range of w either way.
model_mod <- "
  m ~ x + w + x:w
  y ~ m + w
"
res <- mediation_mbco(model_mod, data = d_mod, x = "x", y = "y",
                      moderator = "w", ci_method = "wald",
                      probe_values = c(low = -1, high = 1))

# The band draws B coefficient vectors from their joint normal
# approximation. B = 2000 keeps the example quick; a reported figure
# deserves the default B = 10000. The probed values are marked on
# the curve, and the rug shows where w was observed.
plot_mediation_mbco(res, B = 2000, seed = 113)

# The same curve over a chosen range of w, with a 90% band. The
# 'effects' argument names the pathway to draw; in a model that also
# carries a direct path the total effect is moderated too, and it is
# drawn as a second curve unless 'effects' selects one of them.
plot_mediation_mbco(res, effects = "indirect_via_m", from = -2,
                    to = 2, conf_level = 0.90, B = 2000, seed = 113)
```

---

plot_R2 *Visualize the Proportion of Variance Explained (R^2)*

Description

Creates a horizontal bar chart showing the observed R^2 as a proportion of total variance, with an optional confidence interval displayed beneath the bar and sample size / predictor-count annotations.

Usage

```
plot_R2(
  R2,
  N = NULL,
  p = NULL,
  conf_level = 0.95,
  show_ci = TRUE,
  show_n = TRUE,
  random_predictors = TRUE,
  title = NULL,
  palette = "okabe_ito",
  colors = NULL
)
```

Arguments

R2	The observed squared multiple correlation coefficient ($0 \leq R^2 \leq 1$).
N	Total sample size.
p	Number of predictors.
conf_level	Confidence level for the confidence interval (default 0.95).
show_ci	Logical. If TRUE (the default), a confidence interval is shown beneath the proportion bar. Requires both N and p.
show_n	Logical. If TRUE (the default), N and p are annotated on the plot.
random_predictors	Logical. Whether the predictors are random (TRUE, the default) or fixed. Passed to ci_R2.
title	Optional plot title.
palette	Character string naming the color palette used for the "Explained" portion of the bar when colors is NULL. Defaults to "okabe_ito", base R's colorblind-safe Okabe-Ito palette; "tableau" is also available.
colors	Optional character vector of length 2: the first color fills the "Explained" portion of the bar, the second the "Unexplained" portion. When NULL (the default), the "Explained" portion uses the first color of palette and the "Unexplained" portion a neutral light gray.

Value

A ggplot2 object.

Note

Requires **ggplot2** (listed in Suggests).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*, 524–555. doi:10.1080/00273170802490632

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on R^2 as a model comparison effect size.)

See Also

ci_R2, ci_R, plot_ci, plot_smd

Other plotting: plot_cfa_k(), plot_ci(), plot_equivalence(), plot_forest(), plot_irt_information(), plot_mediation_mbco(), plot_randomization_test(), plot_regions_of_significance(), plot_smd(), plot_trajectories(), plot_trajectories_fitted(), power_equivalence_md_plot()

Examples

```
# The default display: the observed R2 fills its share of the bar, the
# confidence interval sits beneath it, and N and p are annotated.
plot_R2(R2 = 0.25, N = 100, p = 5)

# With fixed predictors and a 90% confidence interval. The interval
# beneath the bar is now the fixed predictor interval from ci_R2, and
# its label reads 90% rather than 95%.
plot_R2(R2 = 0.35, N = 200, p = 3, conf_level = 0.90,
        random_predictors = FALSE)

# Without the interval or the sample size annotation, the bar stands
# alone, which is the display to use when N and p are not known.
plot_R2(R2 = 0.10, show_ci = FALSE, show_n = FALSE)
```

---

plot_randomization_test

Plot the Randomization Distribution Behind a Randomization Test

---

Description

Displays the reference distribution that `randomization_test` built by reassigning the observed scores to the two groups, with the observed statistic marked and every reassignment at least as extreme as the observed one shaded. The shaded proportion is the *p*-value, so the figure shows where that number came from instead of only reporting it.

Usage

```
plot_randomization_test(object, bins = 40L, palette = "okabe_ito", ...)
```

Arguments

`object`	A result of `randomization_test`.
`bins`	Number of histogram bins used to display the reference distribution. Defaults to `40`.
`palette`	Character; the color palette. Defaults to `"okabe_ito"`, base R's colorblind-safe Okabe-Ito palette; `"tableau"` is also available.
`...`	Currently unused; present so the signature can grow without breaking existing calls.

Details

Reading the figure is the point of it. The spread of the distribution is what the reassignments alone can produce when the grouping is irrelevant, which is the null hypothesis of the test. If the observed statistic sits inside that spread, reassignment alone explains it. If it sits out in a tail, few reassignments reproduce it, and that scarcity is the evidence. No normal or *t* distribution appears anywhere in the construction. This is the display Chapter 1 of Maxwell, Delaney, and Kelley (2027) uses to introduce the logic of the randomization test.

Value

A ggplot object, which can be printed or further modified with the usual **ggplot2** verbs.

Author(s)

Ken Kelley

References

Fisher, R. A. (1935). *The design of experiments*. Oliver & Boyd.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 1 on the logic of the randomization test.)

See Also

randomization_test for the test itself.

Other plotting: plot_R2(), plot_cfa_k(), plot_ci(), plot_equivalence(), plot_forest(), plot_irt_information(), plot_mediation_mbco(), plot_regions_of_significance(), plot_smd(), plot_trajectories(), plot_trajectories_fitted(), power_equivalence_md_plot()

Examples

```
treatment <- c(80, 84, 79, 88, 83)
control   <- c(72, 75, 68, 81, 74)
rt <- randomization_test(group_1 = treatment, group_2 = control)
plot_randomization_test(rt)
```

---

plot_regions_of_significance

Plot Regions of Significance for a Covariate by Group Interaction

---

Description

Draws the estimated group difference $\hat{D}(x)$ across the observed range of the covariate, with the confidence band that the region of significance is read from, a reference line at zero, and vertical lines at the boundaries of the region. Wherever the band clears zero the groups differ significantly, so the boundaries are exactly the covariate values at which the band touches the zero line: the plot *is* the decision rule, which is what makes it the natural report of the analysis.

Usage

```
plot_regions_of_significance(
  x,
  data = NULL,
  conf_level = 0.95,
  method = c("simultaneous", "pointwise"),
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  palette = "okabe_ito",
  facet = NULL,
  n_points = 200L
)
```

Arguments

x	A result of regions_of_significance. Alternatively a fitted lm or aov, or a formula with data supplied, in which case regions_of_significance is called first with conf_level and method.
data, conf_level, method	Passed to regions_of_significance, and used only when x is a model or a formula rather than an already computed result.
xlab, ylab	Axis labels. The defaults name the covariate and the group difference.
title	Optional plot title.
palette	Character string naming the color palette. Defaults to "okabe_ito", base R's colorblind-safe Okabe-Ito palette; "tableau" is also available.
facet	Logical. Draw one panel per pair of groups. Defaults to TRUE when there is more than one pair, which keeps the panels from overplotting each other; set it to FALSE to lay the pairs over one another in a single panel.
n_points	Number of covariate values at which the difference and its band are evaluated. Default 200.

Details

The band is $\hat{D}(x) \pm t_{crit}\sqrt{\mathrm{Var}[\hat{D}(x)]}$ with the same critical value used to find the boundaries, so the picture and the table can never disagree. With the default simultaneous critical value (Potthoff, 1964) the band is a simultaneous band: it holds over the whole covariate range at once, which is what licenses scanning it for the covariate values where the groups differ.

The band is drawn over the covariate values actually observed in the two groups. A boundary that falls outside that range is therefore not drawn, deliberately: it is an extrapolation of two fitted lines into a region with no data, and drawing it would invite reading it as a place where something was observed.

Value

A ggplot object. Requires **ggplot2** to be installed.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Johnson, P. O., & Neyman, J. (1936). Tests of certain linear hypotheses and their application to some educational problems. *Statistical Research Memoirs, 1*, 57–93.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9 and its extension on heterogeneity of regression.)

Potthoff, R. F. (1964). On the Johnson-Neyman technique and some extensions thereof. *Psychometrika, 29*(3), 241–256. doi:10.1007/BF02289721

See Also

regions_of_significance, plot_ci

Other plotting: plot_R2(), plot_cfa_k(), plot_ci(), plot_equivalence(), plot_forest(), plot_irt_information(), plot_mediation_mbco(), plot_randomization_test(), plot_smd(), plot_trajectories(), plot_trajectories_fitted(), power_equivalence_md_plot()

Examples

```
# The Pygmalion teacher-expectancy data: post-test IQ (averaged over
# the two follow-ups) on pretest IQ, by condition. The expectancy
# effect is significant only in a band of pretest IQ values.
data(pygmalion)
pygmalion$iq_post <- (pygmalion$iq_4 + pygmalion$iq_8) / 2
fit <- lm(iq_post ~ iq_pre * treatment, data = pygmalion)
plot_regions_of_significance(fit)

# Three groups: one panel per pair.
set.seed(113)
n <- 150
g <- factor(rep(c("control", "low", "high"), each = n / 3))
x <- rnorm(n, 50, 10)
y <- 2 + 0.5 * x + (g == "high") * (0.4 * x - 15) + rnorm(n, 0, 5)
plot_regions_of_significance(y ~ x * g, data = data.frame(y, x, g))
```

---

plot_smd *Visualize a Standardized Mean Difference With Overlapping Distributions*

Description

Creates a publication-quality plot showing two normal distributions separated by the standardized mean difference (*d*). The plot includes a confidence interval for the population effect size and sample size annotations, both shown by default.

Usage

```
plot_smd(
  smd = NULL,
  n_1 = NULL,
  n_2 = NULL,
  group_1 = NULL,
  group_2 = NULL,
  conf_level = 0.95,
  show_ci = TRUE,
  show_n = TRUE,
  title = NULL,
  group_labels = c("Group 1", "Group 2"),
  palette = "okabe_ito",
  colors = NULL
)
```

Arguments

smd	The standardized mean difference (Cohen's *d*).
n_1	Sample size for Group 1.
n_2	Sample size for Group 2.
group_1	Raw data for Group 1. When provided, smd, n_1, and n_2 are computed from the data.
group_2	Raw data for Group 2.
conf_level	Confidence level for the confidence interval (default 0.95).
show_ci	Logical. If TRUE (the default), a confidence interval for the population standardized mean difference is displayed beneath the distributions. Requires both n_1 and n_2.
show_n	Logical. If TRUE (the default), per-group sample sizes are annotated on the plot.
title	Optional character string for the plot title. Defaults to "Standardized Mean Difference".
group_labels	Character vector of length 2 giving labels for the two groups. Defaults to c("Group 1", "Group 2").
palette	Character string naming the color palette used when colors is NULL. Defaults to "okabe_ito", base R's colorblind-safe Okabe-Ito palette; "tableau" is also available.
colors	Optional character vector of length 2 giving fill colors for the two groups. When NULL (the default), the first two colors of palette are used.

Details

Two unit-variance normal distributions are drawn, centered at 0 (Group 2 / reference) and *d* (Group 1 / focal). The semi-transparent fills make the overlap visible, giving a direct visual impression of how much the distributions differ.

When show_ci = TRUE and both n_1 and n_2 are available, the function calls ci_smd to compute the noncentral *t* based confidence interval and displays it as a horizontal bar beneath the curves. A filled dot marks the point estimate and vertical caps mark the confidence bounds.

Value

A ggplot2 object that can be further customized with standard **ggplot2** layers, scales, and themes.

Note

Requires **ggplot2** (listed in Suggests). Install it with `install.packages("ggplot2")` if needed.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

See Also

smd, ci_smd, plot_ci, plot_R2

Other plotting: plot_R2(), plot_cfa_k(), plot_ci(), plot_equivalence(), plot_forest(), plot_irt_information(), plot_mediation_mbco(), plot_randomization_test(), plot_regions_of_significance(plot_trajectories(), plot_trajectories_fitted(), power_equivalence_md_plot()

Other confidence intervals for effect sizes: ci_R2(), ci_c(), ci_c_ancova(), ci_c_ancova_bp(), ci_correlation, ci_cv(), ci_eta_squared(), ci_eta_squared_generalized(), ci_eta_squared_partial(), ci_mahalanobis(), ci_omega_squared(), ci_pvaf(), ci_rc(), ci_reg_coef(), ci_rmsea(), ci_sc(), ci_sc_ancova(), ci_sm(), ci_smd(), ci_smd_c(), ci_snr(), ci_src(), ci_srsnr(), contrast_adjusted()

Examples

```
# From a known standardized mean difference and the two sample sizes:
# the two curves, the interval beneath them, and the sample sizes in
# the corner are the default display.
plot_smd(smd = 0.50, n_1 = 50, n_2 = 50)

# From raw data, where the standardized mean difference and both
# sample sizes are taken from the data. The d annotation now reports
# the sample value rather than a supplied one.
set.seed(113)
g1 <- rnorm(40, mean = 0.6, sd = 1)
g2 <- rnorm(40, mean = 0.0, sd = 1)
plot_smd(group_1 = g1, group_2 = g2)
```

```
# Without the confidence interval or the sample size annotations, the
# figure is the two curves alone.
plot_smd(smd = 0.80, show_ci = FALSE, show_n = FALSE)

# Group labels and a title of the reader's own.
plot_smd(smd = 0.45, n_1 = 75, n_2 = 75,
         group_labels = c("Treatment", "Control"),
         title = "Treatment Effect on Reading Scores")
```

`plot_trajectories` *Visualize Observed Individual Trajectories in a Longitudinal Data Set*

Description

Plots one trajectory per subject from a long-format data frame, optionally colored by a grouping variable, and optionally faceted into one panel per subject. Returns a **ggplot2** object that can be further customized.

Usage

```
plot_trajectories(
  data,
  id,
  time,
  outcome,
  group = NULL,
  ids = NULL,
  n_random = NULL,
  pct_random = NULL,
  facet = FALSE,
  nrow = NULL,
  ncol = NULL,
  show_points = TRUE,
  point_size = 1.5,
  linewidth = 0.5,
  alpha = 0.7,
  palette = "okabe_ito",
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  seed = NULL
)
```

Arguments

`data`	A long-format `data.frame` (one row per subject-occasion).
`id`	Character. Column name in `data` identifying the subject.

time	Character. Column name for the time / occasion variable (the *x* axis).
outcome	Character. Column name for the outcome / score variable (the *y* axis).
group	Optional character. Column name for a grouping variable used to color the trajectories (and panels, if faceted).
ids	Optional vector of subject IDs to plot.
n_random	Optional integer; randomly sample this many subjects.
pct_random	Optional numeric; sample this percentage of subjects. Values ≤ 1 are interpreted as proportions; values > 1 as percentages. At most one of ids, n_random, or pct_random may be supplied.
facet	Logical. If TRUE, draw one panel per subject via facet_wrap. If FALSE (default), overlay all trajectories in one panel.
nrow, ncol	Optional integers passed to facet_wrap() when facet = TRUE.
show_points	Logical. If TRUE (default), draw the observed points as well as the connecting lines.
point_size	Size of the observed points (default 1.5).
linewidth	Line width for the connecting segments (default 0.5).
alpha	Transparency for points and lines (default 0.7).
palette	Character string naming the color palette used to color the trajectories when group is a discrete (factor, character, or logical) variable. Defaults to "okabe_ito", base R's colorblind-safe Okabe-Ito palette; "tableau" is also available. Ignored when group is NULL or numeric.
title, xlab, ylab	Optional plot labels. Sensible defaults are taken from outcome and time when these are NULL.
seed	Optional integer random seed used when n_random or pct_random is supplied. Defaults to NULL, which leaves the user's current RNG state intact; supply an integer for reproducible subject sampling.

Details

The function modernizes the original vit() (visualize individual trajectories) function by returning a single **ggplot2** object instead of producing graphical side effects. Saving is handled by the user via ggsave; multi-page output via faceting and facet_wrap's nrow/ncol.

Value

A ggplot object.

Note

Requires **ggplot2** (a Suggests dependency).

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

plot_trajectories_fitted for plotting observed trajectories together with a fitted multilevel model's predictions.

Other plotting: plot_R2(), plot_cfa_k(), plot_ci(), plot_equivalence(), plot_forest(), plot_irt_information(), plot_mediation_mbco(), plot_randomization_test(), plot_regions_of_significance(plot_smd(), plot_trajectories_fitted(), power_equivalence_md_plot()

Examples

```
# The Orthodont data from nlme: 27 children, 4 measurements each.
d <- nlme::Orthodont

# Overlay all trajectories, colored by sex.
plot_trajectories(d, id = "Subject", time = "age",
                  outcome = "distance", group = "Sex")

# One panel per child, for twelve children drawn at random. The seed
# makes the draw reproducible, and the session's generator state is
# left as it was.
plot_trajectories(d, id = "Subject", time = "age",
                  outcome = "distance",
                  n_random = 12, facet = TRUE, ncol = 4,
                  seed = 113)
```

---

plot_trajectories_fitted

Plot Observed and Fitted Individual Trajectories From a Multilevel Model

---

Description

Given a fitted lme/nlme (**nlme**) or lmer (**lme4**) model, plots each subject's observed values and fitted curve on a smooth time grid, faceted one panel per subject. Per-subject R^2 (squared correlation between observed and fitted values) and root-mean-square error are computed and attached to the returned **ggplot2** object as the quality_of_fit attribute.

Usage

```
plot_trajectories_fitted(
  model,
  id = NULL,
  time = NULL,
  outcome = NULL,
  ids = NULL,
  n_random = NULL,
  pct_random = NULL,
  n_grid = 100,
```

```
  show_points = TRUE,
  point_size = 1.5,
  linewidth = 0.6,
  alpha = 0.8,
  palette = "okabe_ito",
  nrow = NULL,
  ncol = NULL,
  show_quality = TRUE,
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  seed = NULL
)
```

Arguments

`model`	A fitted model object of class `lme`, `nlme`, or `lmerMod`.
`id`, `time`, `outcome`	Optional character names of the ID, time, and outcome columns. When `NULL` (the default), each is auto-detected from the model object: the outcome is the response variable in the formula, the ID is the first random-effect grouping factor, and time is the first fixed-effect predictor. Override these when the auto-detection is wrong.
`ids`, `n_random`, `pct_random`	Subject-subsetting options identical to those of `plot_trajectories`; at most one may be supplied.
`n_grid`	Integer. Number of points used to draw each subject's smooth fitted curve (default `100`).
`show_points`	Logical. Whether to draw the observed values (default `TRUE`).
`point_size`, `linewidth`, `alpha`, `nrow`, `ncol`	Visual / layout controls.
`palette`	Character string naming the color palette; the fitted curve is drawn in the palette's primary color. Defaults to `"okabe_ito"`, base R's colorblind-safe Okabe-Ito palette; `"tableau"` is also available.
`show_quality`	Logical. If `TRUE` (the default), each panel strip text includes the subject's R^2 and RMSE.
`title`, `xlab`, `ylab`	Optional plot labels.
`seed`	Optional integer random seed used when `n_random` or `pct_random` is supplied. Defaults to `NULL`, which leaves the user's current RNG state intact; supply an integer for reproducible subject sampling.

Details

Modernizes the original `vit_fitted()` function by:

- returning a **ggplot2** object instead of writing to graphics devices,

- attaching per-subject quality-of-fit as an attribute rather than assigning it into the global environment, a serious side effect of the original,
- drawing a smooth fitted curve from a per-subject time grid via `predict(..., re.form = NULL)` for **lme4** fits and `predict(..., level = 1)` for **nlme** fits,
- correctly identifying **lme4** fits (which use class `lmerMod`, not `lmer`).

Value

A `ggplot` object. The per-subject quality-of-fit `data.frame` (columns: id column, `r_squared`, `rmse`) is attached as `attr(<plot>, "quality_of_fit")`.

Note

Requires **ggplot2** plus, depending on the model class, **nlme** or **lme4** (`Suggests` dependencies).

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

`plot_trajectories`

Other plotting: `plot_R2()`, `plot_cfa_k()`, `plot_ci()`, `plot_equivalence()`, `plot_forest()`, `plot_irt_information()`, `plot_mediation_mbco()`, `plot_randomization_test()`, `plot_regions_of_significance(`
`plot_smd()`, `plot_trajectories()`, `power_equivalence_md_plot()`

Other within-subjects analysis: `anova_within()`, `anova_within_two_way()`, `epsilon_corrections()`, `mauchly_test()`, `pairwise_within()`

Examples

```
# nlme: linear growth in tooth distance over age for the 27 Orthodont
# children. Four of the children are paneled here so the figure is
# quick to draw; drop n_random to get a panel for every child.
fm_nlme <- nlme::lme(distance ~ age, random = ~ age | Subject,
                     data = nlme::Orthodont)
p <- plot_trajectories_fitted(fm_nlme, n_random = 4, seed = 113)
p
attr(p, "quality_of_fit")  # per-subject R^2 and RMSE

# An lme4 fit is handled the same way: the outcome, the subject
# identifier, and the time variable are read from the model. Six of
# the eighteen sleepstudy subjects are paneled here.
fm_lme4 <- lme4::lmer(Reaction ~ Days + (Days | Subject),
                      data = lme4::sleepstudy)
plot_trajectories_fitted(fm_lme4, n_random = 6, seed = 113)
```

power_density_equivalence_md *Density Underlying the TOST Power Calculation*

Description

Evaluates the integrand whose integral over $(0, \text{upper})$ yields the power of the Schuirmann (1987) two one-sided tests procedure; see power_equivalence_md. Useful for plotting the integrand and for diagnostic work.

Usage

```
power_density_equivalence_md(
  power_sigma,
  alpha_level,
  theta1,
  theta2,
  diff,
  sigma,
  n,
  nu
)
```

Arguments

power_sigma	Numeric vector of σ values at which to evaluate the integrand.
alpha_level	Type I error rate for each of the two one-sided tests.
theta1	Lower limit of the equivalence interval on the appropriate scale (regular or log).
theta2	Upper limit of the equivalence interval on the appropriate scale (regular or log).
diff	True difference in treatment means (ratio on the log scale) on the appropriate scale.
sigma	$\sqrt{\text{error variance}}$.
n	Number of subjects per treatment.
nu	Degrees of freedom for sigma.

Value

A data.frame with one row per supplied power_sigma, and columns power_sigma and power_density.

Note

See the legacy MBESS package (Kelley, 2007a, 2007b) for additional details and discussion.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007a). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K. (2007b). Methods for the behavioral, educational, and social sciences: An R package. *Behavior Research Methods, 39*(4), 979–984. doi:10.3758/BF03192993

Phillips, K. F. (1990). Power of the two one-sided tests procedure in bioequivalence. *Journal of Pharmacokinetics and Biopharmaceutics, 18*(2), 139–144. doi:10.1007/BF01063556

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics, 15*(6), 657–680.

See Also

power_equivalence_md, power_equivalence_md_plot

Other equivalence testing: equivalence_c(), equivalence_r(), equivalence_smd(), plot_equivalence(), power_equivalence_c(), power_equivalence_md(), power_equivalence_md_plot(), ss_power_equivalence_c()

Examples

```
# Density at a single value of sigma:
power_density_equivalence_md(power_sigma = 0.10, alpha_level = .05,
                             theta1 = -.2, theta2 = .2, diff = .05,
                             sigma = .20, n = 24, nu = 22)

# Vectorized over a grid:
grid <- power_density_equivalence_md(
  power_sigma = seq(0.01, 0.40, length.out = 50),
  alpha_level = .05, theta1 = -.2, theta2 = .2, diff = .05,
  sigma = .20, n = 24, nu = 22
)
head(grid)
```

---

power_equivalence_c *Power of the TOST or Noninferiority Test for a Linear Contrast*

Description

Computes the exact power of the Schuirmann (1987) two one-sided tests procedure, or of the one-sided noninferiority test, for a linear contrast of group means $\psi = \sum_j c_j \mu_j$ with one pooled error term. For equivalence, the power is the probability that the $(1 - 2\alpha)$ confidence interval for ψ lies entirely inside $(-\delta_L, \delta_U)$, computed by numerical integration over the chi distribution of the estimated error standard deviation; for noninferiority, the power is a noncentral t probability in closed form. This is the contrast generalization of power_equivalence_md.

Usage

```
power_equivalence_c(
  c_weights,
  n,
  sigma,
  delta_lower = NULL,
  delta_upper = NULL,
  true_psi = 0,
  alpha_level = 0.05,
  side = c("equivalence", "noninferiority"),
  df_error = NULL
)
```

Arguments

`c_weights`	The contrast weights. The weights must sum to zero with the positive weights summing to 1 and the negative weights to -1, so that the bounds are on the raw scale of the response.
`n`	Sample sizes per group (if length 1, equal group sizes are assumed). Together with `c_weights`, `n` determines the standard error factor $\sqrt{\sum_j c_j^2/n_j}$.
`sigma`	The error standard deviation (the square root of the mean square error).
`delta_lower, delta_upper`	Equivalence bounds on the raw scale of the response. Both must be positive; the equivalence region is $(-\delta_L, +\delta_U)$. If only `delta_upper` is supplied, the bounds are symmetric. Noninferiority uses $-\delta_L$ alone.
`true_psi`	The population value of the contrast at which the power is evaluated. Default `0`, the most favorable point for an equivalence declaration.
`alpha_level`	One-sided significance level for each test. Default `0.05`.
`side`	`"equivalence"` (default) for the TOST power, or `"noninferiority"` for the one-sided test against $-\delta_L$.
`df_error`	The error degrees of freedom. Defaults to $N - J$; supply it directly when the error term comes from a model with more groups or additional predictors than the contrast involves (for example, a five-group model supplying the pooled error for a two-group contrast).

Details

Equivalence power. Conditional on the estimated error standard deviation S, the $(1 - 2\alpha)$ CI fits inside the bounds on a computable event, and the unconditional power integrates that event over the scaled chi distribution of S on `df_error` degrees of freedom. With `c_weights = c(1, -1)` and equal `n`, the result reproduces `power_equivalence_md` exactly.

Noninferiority power. The one-sided test rejects when $t = (\hat{\psi} + \delta_L)/\mathrm{SE}(\hat{\psi})$ exceeds $t_{1-\alpha,\nu}$, so the power is $\Pr(T'_\nu(\lambda) > t_{1-\alpha,\nu})$ with noncentrality $\lambda = (\psi + \delta_L)/(\sigma\sqrt{\sum_j c_j^2/n_j})$.

The feasibility condition. If the expected half-width of the CI is not smaller than the bounds allow, the equivalence power is zero or near zero regardless of `true_psi`: an imprecise design cannot

declare equivalence even when the arms are truly identical. Planning should target a half-width of about half the bound; see ss_power_equivalence_c and ss_aipe_c.

Value

A one-row data.frame with columns term ("power") and value (the computed power, in $[0, 1]$).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Chattopadhyay, B., Bandyopadhyay, T., Kelley, K., & Padalunkal, J. J. (2025). A sequential approach for noninferiority or equivalence of a linear contrast under cost constraints. *Psychological Methods, 30*(2), 425–439. doi:10.1037/met0000570

Phillips, K. F. (1990). Power of the two one-sided tests procedure in bioequivalence. *Journal of Pharmacokinetics and Biopharmaceutics, 18*(2), 139–144. doi:10.1007/BF01063556

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics, 15*(6), 657–680.

See Also

power_equivalence_md, ss_power_equivalence_c, equivalence_c, ss_aipe_c

Other equivalence testing: equivalence_c(), equivalence_r(), equivalence_smd(), plot_equivalence(), power_density_equivalence_md(), power_equivalence_md(), power_equivalence_md_plot(), ss_power_equivalence_c()

Examples

```
# 1. Two groups of 61 and 113 sharing a five-group pooled error term
#    (so df_error = 404 - 5 = 399), bounds of 5 raw-scale points:
#    the design's probability of declaring equivalence when the
#    groups are truly identical.
power_equivalence_c(c_weights = c(1, -1), n = c(61, 113),
                    sigma = 15.67, delta_upper = 5,
                    true_psi = 0, df_error = 399)

# 2. The same design's noninferiority power at the same point.
power_equivalence_c(c_weights = c(1, -1), n = c(61, 113),
                    sigma = 15.67, delta_upper = 5,
                    true_psi = 0, df_error = 399,
                    side = "noninferiority")

# 3. Agreement with power_equivalence_md() in the two-group case
#    (Phillips, 1990, Table 1: expected 0.8029678).
power_equivalence_c(c_weights = c(1, -1), n = 24, sigma = 0.20,
                    delta_lower = 0.2, delta_upper = 0.2,
                    true_psi = 0.05, df_error = 22)
```

power_equivalence_md *Power of the Two One-Sided Tests Procedure (TOST) for Equivalence*

Description

Computes the power of the Schuirmann (1987) two one-sided tests procedure , the probability that a $(1-2\alpha)$ confidence interval for the mean difference (or ratio, on the log scale) lies entirely within the equivalence interval $[\theta_1, \theta_2]$, by numerical integration over the chi distribution of the sample standard deviation.

Usage

```
power_equivalence_md(
  alpha_level,
  logscale,
  ltheta1,
  ltheta2,
  ldiff,
  sigma,
  n,
  nu
)
```

Arguments

alpha_level	Type I error rate for each of the two one-sided tests (typically 0.05). The full equivalence test uses a $(1-2\alpha)$ confidence interval.
logscale	Logical. If TRUE, treatment means are compared on the logarithmic scale; ltheta1, ltheta2, and ldiff are expected as ratios (untransformed) and are log-transformed internally.
ltheta1	Lower limit of the equivalence interval (on the original scale; logged internally if logscale = TRUE).
ltheta2	Upper limit of the equivalence interval (on the original scale; logged internally if logscale = TRUE).
ldiff	True difference in treatment means (or ratio on the log scale).
sigma	$\sqrt{\text{error variance}}$; root-MSE from an ANOVA. On the log scale, this is the coefficient of variation.
n	Number of subjects per treatment (or total subjects in a crossover design).
nu	Degrees of freedom associated with sigma.

Details

The computation conditions on the error standard deviation the study will actually observe. Given that value, whether the confidence interval fits inside the equivalence interval is an ordinary normal probability, and the power is that probability averaged over the chi distribution the error standard

deviation follows on nu degrees of freedom. The averaging is carried out on a unit-free scale, with the equivalence limits expressed in standard errors and the error standard deviation as a multiple of sigma, so the power depends on the design rather than on the units of the response: multiplying ltheta1, ltheta2, ldiff, and sigma by a common factor leaves the answer unchanged.

For Phillips's (1990) original example (regular-scale two-period crossover with $\theta_1 = -0.2$, $\theta_2 = 0.2$, CV $= 0.20$, $\delta = 0.05$, $n = 24$, $\nu = 22$), this function reproduces the published value of 0.8029678 (Phillips, 1990, Table 1, 5th row, 5th column).

Value

A one-row data.frame with columns term ("power") and value (the computed power, in $[0, 1]$).

Note

See the legacy MBESS package (Kelley, 2007a, 2007b) for additional details and discussion.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Diletti, E., Hauschke, D., & Steinijans, V. W. (1991). Sample size determination of bioequivalence assessment by means of confidence intervals. *International Journal of Clinical Pharmacology, Therapy and Toxicology, 29*(1), 1–8.

Kelley, K. (2007a). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K. (2007b). Methods for the behavioral, educational, and social sciences: An R package. *Behavior Research Methods, 39*(4), 979–984. doi:10.3758/BF03192993

Phillips, K. F. (1990). Power of the two one-sided tests procedure in bioequivalence. *Journal of Pharmacokinetics and Biopharmaceutics, 18*(2), 139–144. doi:10.1007/BF01063556

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics, 15*(6), 657–680.

See Also

power_equivalence_md_plot, power_density_equivalence_md

Other equivalence testing: equivalence_c(), equivalence_r(), equivalence_smd(), plot_equivalence(), power_density_equivalence_md(), power_equivalence_c(), power_equivalence_md_plot(), ss_power_equivalence_c()

Examples

```
# Table 1 of Phillips, 1990, fifth row and fifth column, where the
# published power is 0.8029678.
power_equivalence_md(alpha_level = .05, logscale = FALSE,
                     ltheta1 = -.2, ltheta2 = .2, ldiff = .05,
```

```
                    sigma = .20, n = 24, nu = 22)

# Table 1 of Diletti et al., 1991, on the log scale, so the limits and
# the true difference are ratios of test to reference. The published
# power is 0.7922796.
power_equivalence_md(alpha_level = .05, logscale = TRUE,
                    ltheta1 = .8, ltheta2 = 1.25, ldiff = 1.05,
                    sigma = .20, n = 18, nu = 16)
```

`power_equivalence_md_plot` *Plot TOST Equivalence-Test Power Curves Over a Range of True Differences*

Description

For each sample size in `n`, draws power as a function of the true mean difference (or ratio, on the log scale), evaluated at 201 equally spaced points across the equivalence interval. Returns a **ggplot2** object; the underlying numerical grid is attached as `attr(<plot>, "power_grid")`.

Usage

```
power_equivalence_md_plot(
  alpha_level,
  logscale,
  theta1,
  theta2,
  sigma,
  n,
  nu,
  title = NULL,
  subtitle = NULL
)
```

Arguments

`alpha_level`	Type I error rate for each of the two one-sided tests.
`logscale`	Logical. If `TRUE`, the means are compared on the logarithmic scale.
`theta1`	Lower limit of the equivalence interval.
`theta2`	Upper limit of the equivalence interval.
`sigma`	$\sqrt{\text{error variance}}$.
`n`	Vector of sample sizes (one curve per element).
`nu`	Vector of degrees of freedom for `sigma`, the same length as `n`.
`title`	Optional plot title (default `"Power of TOST"`).
`subtitle`	Optional subtitle (typically a reference like `"Phillips, Figure 3"`).

Value

A ggplot object. The 201-row power grid (column 1: true difference; remaining columns: power for each n) is attached as attr(<plot>, "power_grid").

Note

See the legacy MBESS package (Kelley, 2007a, 2007b) for additional details and discussion.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Diletti, E., Hauschke, D., & Steinijans, V. W. (1991). Sample size determination of bioequivalence assessment by means of confidence intervals. *International Journal of Clinical Pharmacology, Therapy and Toxicology, 29*(1), 1–8.

Kelley, K. (2007a). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K. (2007b). Methods for the behavioral, educational, and social sciences: An R package. *Behavior Research Methods, 39*(4), 979–984. doi:10.3758/BF03192993

Phillips, K. F. (1990). Power of the two one-sided tests procedure in bioequivalence. *Journal of Pharmacokinetics and Biopharmaceutics, 18*(2), 139–144. doi:10.1007/BF01063556

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics, 15*(6), 657–680.

See Also

power_equivalence_md, power_density_equivalence_md

Other equivalence testing: equivalence_c(), equivalence_r(), equivalence_smd(), plot_equivalence(), power_density_equivalence_md(), power_equivalence_c(), power_equivalence_md(), ss_power_equivalence_c()

Other plotting: plot_R2(), plot_cfa_k(), plot_ci(), plot_equivalence(), plot_forest(), plot_irt_information(), plot_mediation_mbco(), plot_randomization_test(), plot_regions_of_significance(plot_smd(), plot_trajectories(), plot_trajectories_fitted()

Examples

```
# One curve per sample size, showing power against the true mean
# difference. The seven sample sizes are the ones behind Figure 3 of
# Phillips, 1990, so the figure reproduces that one; every curve
# evaluates the power integral at 201 true differences.
n  <- c(9, 12, 18, 24, 30, 40, 60)
nu <- c(7, 10, 16, 22, 28, 38, 58)
fig <- power_equivalence_md_plot(
  alpha_level = .05, logscale = FALSE,
  theta1 = -.2, theta2 = .2, sigma = .20,
  n = n, nu = nu,
```

```
  subtitle = "Phillips Figure 3"
)
fig

# The numbers behind the curves travel with the figure, so a particular
# power value can be read off rather than eyeballed. The first column is
# the true difference and the remaining columns give power, one column
# per sample size. Power is highest where the true difference is zero.
power_grid <- attr(fig, "power_grid")
power_grid[which.min(abs(power_grid[, 1])), ]

# Figure 1c of Diletti et al., 1991, is the same idea on the log scale,
# where the equivalence limits are the 0.80 to 1.25 ratio bounds used
# in bioequivalence work.
n_d  <- c(8, 12, 18, 24, 30, 40, 60)
nu_d <- c(6, 10, 16, 22, 28, 38, 58)
power_equivalence_md_plot(
  alpha_level = .05, logscale = TRUE,
  theta1 = .8, theta2 = 1.25, sigma = .20,
  n = n_d, nu = nu_d,
  subtitle = "Diletti, Figure 1c"
)
```

---

power_fisher_exact *Power of Fisher's Exact Test (Noncentral Hypergeometric)*

Description

Computes the power of Fisher's exact test (Fisher, 1934) for the 2×2 table under Fisher's noncentral hypergeometric distribution, where the alternative is parameterized by the true odds ratio ψ. The power is the probability that the (conditional) exact test rejects $H_0 : \psi = 1$ when in fact $\psi = \psi_1 \neq 1$.

Usage

```
power_fisher_exact(
  n_1,
  n_2,
  p_1,
  p_2,
  alpha_level = 0.05,
  alternative = c("two_sided", "less", "greater")
)
```

Arguments

n_1, n_2	Group sample sizes for the two columns of the 2×2 table (e.g., treatment vs control).

p_1, p_2	Success probabilities in the two groups under the alternative. The odds ratio under the alternative is $\psi = (p_1/(1-p_1))/(p_2/(1-p_2))$.
alpha_level	Two-sided significance level. Default 0.05.
alternative	One of "two_sided" (default; the base-R spelling "two.sided" is accepted as an alias), "less", or "greater".

Details

Setup. Fisher's exact test conditions on the marginal totals of the 2×2 table:

	Success	Failure	Total
Group 1	X_1	$n_1 - X_1$	n_1
Group 2	$S - X_1$	$(n_1 + n_2) - n_1 - S + X_1$	n_2
Total	S	$n_1 + n_2 - S$	$n_1 + n_2$

Under $H_0 : \psi = 1$, X_1 given the marginals follows the *central* hypergeometric. Under the alternative ψ_1, X_1 follows *Fisher's noncentral* hypergeometric with odds ratio parameter ψ_1 (Fisher, 1935; Fog, 2008), the conditional distribution of one binomial count given the total of two independent binomials. (Wallenius' noncentral hypergeometric, which arises from sequential biased urn sampling, is a different distribution and is not the relevant one here.)

Power calculation. For each possible value of the column-1 total $S = 0, 1, \ldots, n_1 + n_2$:

1. Determine the rejection region under $H_0 : \psi = 1$ using the central hypergeometric.
2. Compute $\Pr(X_1 \in \text{reject} \mid \psi = \psi_1, S)$ under the noncentral hypergeometric.
3. Weight by $\Pr(S \mid \psi_1)$, the marginal probability of total column-1 successes under the alternative.

The power is the resulting weighted sum.

Value

A data.frame with rows for the power, the alternative-odds-ratio ψ_1, the alternative success probabilities p_1 and p_2, the expected column-1 total $E[S]$ across both groups, and the row / column totals.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Fisher, R. A. (1934). *Statistical methods for research workers* (5th ed.). Oliver & Boyd.

Fisher, R. A. (1935). The logic of inductive inference. *Journal of the Royal Statistical Society, 98*(1), 39–82.

Fog, A. (2008). Sampling methods for Wallenius' and Fisher's noncentral hypergeometric distributions. *Communications in Statistics – Simulation and Computation, 37*(2), 241–257. doi:10.1080/03610910701790236

Good, P. I. (2000). *Permutation tests: A practical guide to resampling methods for testing hypotheses* (2nd ed.). Springer.

O'Brien, R. G. (1998). A tour of UnifyPow: A SAS module/macro for sample size analysis. *Proceedings of the 23rd SAS Users Group International Conference*, 1346–1355.

See Also

`fisher.test`

Other sample size for power: `ss_aipe_mixed_effects()`, `ss_power_R2()`, `ss_power_R2_sensitivity()`, `ss_power_c()`, `ss_power_c_ancova()`, `ss_power_composite_ancova()`, `ss_power_composite_ancova_2group()`, `ss_power_composite_anova()`, `ss_power_composite_factorial_ancova()`, `ss_power_composite_factorial_ancova`
`ss_power_composite_factorial_anova()`, `ss_power_composite_sem()`, `ss_power_contrast()`, `ss_power_equivalence_c()`, `ss_power_factorial_ancova()`, `ss_power_factorial_anova()`, `ss_power_indirect_effect()`, `ss_power_mixed_effects()`, `ss_power_one_way_anova()`, `ss_power_pcm()`, `ss_power_r()`, `ss_power_rc()`, `ss_power_reg_coef()`, `ss_power_reg_coef_sensitivity()`, `ss_power_rm_anova()`, `ss_power_sc()`, `ss_power_sem()`, `ss_power_smd()`, `ss_power_split_plot_anova()`

Examples

```
# 1. Power for n_1 = n_2 = 30 when the two population proportions are
#    0.6 and 0.3. The value comes from enumerating the conditional
#    reference set the test itself uses, not from a normal
#    approximation, so it is the power of the test as conducted.
power_fisher_exact(n_1 = 30, n_2 = 30, p_1 = 0.6, p_2 = 0.3)

# 2. A difference of 0.10 between proportions is much harder to detect:
#    100 per group is not close to enough. Sample size requirements grow
#    quickly as the difference between the two proportions shrinks.
power_fisher_exact(n_1 = 100, n_2 = 100, p_1 = 0.45, p_2 = 0.35)
```

---

prime_time_achievement

Indiana Prime Time Third Grade Achievement Evaluation Data

Description

The complete student level data file from the 2000 to 2001 Indiana Department of Education program evaluation of Project Prime Time, reported in Lapsley, Daytner, Kelley, and Maxwell (2002, ERIC ED466679). The evaluation examined the academic performance of N = 10,927 third grade students in n = 573 classrooms (here 586 by the `paste(corp, school, class)` rule), n = 163 schools, n = 61 school corporations, and 9 Indiana educational service regions as a function of class size, pupil to teacher ratio, and the presence of a Prime Time instructional assistant. The data have been used as a multilevel example through chapters 3, 4, 6, 9, and 10 of Finch, Bolin, and Kelley (2019, *Multilevel Modeling Using R*, 2nd ed., CRC Press) and are made available here as a benchmark data set for the design, measurement, and analysis of nested data.

Usage

```
prime_time_achievement
```

Format

A data frame with 10,927 observations on 113 variables. Variables fall into seven blocks: three derived unique cluster identifiers, student level demographics and ability and achievement scores, classroom level variables, school level variables, and school corporation (district) level variables. Original Indiana DOE variable spellings are preserved, including the typos `calender` (calendar), `hispanc1` and `hispanc2` (Hispanic), and `rmediate` (remediate), for code compatibility with Finch, Bolin, and Kelley (2019). The SPSS variable label from the source file is available as `attr(prime_time_achievement$VAR, "label")` for every variable carried over from the SPSS file; the one derived recode without a source label is `classize` (see its entry below).

`id` Student identifier in the source file (not guaranteed unique; see `class_id` etc. for stable cluster keys).

`region` Indiana educational service region, coded 1 to 9. Sampling stratifier (25% of corporations per region).

`corp` School corporation (district) numeric identifier. **Note:** corp 2400 appears in both region 2 (12 schools) and region 3 (1 school) in the source file, so `paste(region, corp)` is the cleaner cluster key; see `corp_id`.

`school` Numeric school identifier (unique within corporation).

`class` Classroom number within school, 1 to 8. **Not** unique across schools.

`corp_id` Derived. `paste(region, corp, sep = "_")`. 61 distinct values, matching the count of school corporations reported in Lapsley et al. (2002).

`school_id` Derived. `paste(corp, school, sep = "_")`. 163 distinct values.

`class_id` Derived. `paste(corp, school, class, sep = "_")`. 586 distinct values. (The published report counted 573 classrooms; the small discrepancy reflects a different counting convention used in the manuscript.)

`gender` 1 = Female, 2 = Male. 45 `NA`.

`age` Student age in months.

`race` Indiana DOE 6-category ethno-racial code: 1 = American Indian / Alaskan, 2 = African American, 3 = Asian American, 4 = Hispanic American, 5 = Caucasian American, 6 = Multiracial.

`gread` Gates-MacGinitie reading.

`gevocab` Gates-MacGinitie vocabulary.

`greadcm` Gates-MacGinitie reading composite.

`gelang` Gates-MacGinitie language.

`gelangmc` Gates-MacGinitie language mechanics.

`gelangcm` Gates-MacGinitie language composite.

`gemath` Gates-MacGinitie mathematics.

`gemathcp` Gates-MacGinitie mathematics computation.

`gemathcm` Gates-MacGinitie mathematics composite.

`getotal` Gates-MacGinitie total.

`ncread` NCE reading (ISTEP+).

`ncvocab` NCE vocabulary (ISTEP+).

`ncreadcm` NCE reading composite (ISTEP+).

`nclang` NCE language (ISTEP+).

`nclangmc` NCE language mechanics (ISTEP+).

`nclangcm` NCE language composite (ISTEP+).

`ncmath` NCE mathematics (ISTEP+).

`ncmathcp` NCE mathematics computation (ISTEP+).

`ncmathcm` NCE mathematics composite (ISTEP+).

`nctotal` **NCE total composite (ISTEP+)**, the criterion variable in the Lapsley et al. (2002) HLM analyses.

`aaread` AANCE reading.

`aavocab` AANCE vocabulary.

`aareadcm` AANCE reading composite.

`aalang` AANCE language.

`aalangmc` AANCE language mechanics.

`aalangcm` AANCE language composite.

`aamath` AANCE mathematics.

`aamathcp` AANCE mathematics computation.

`aamathcm` AANCE mathematics composite.

`aatotal` AANCE total.

`npanverb` NPA nonverbal reasoning.

`npamem` NPA working memory.

`npaverb` NPA verbal reasoning.

`npatotal` NPA total.

`csi` Cognitive Skills Index (student level).

`multi` Multi-age classroom indicator (1 = Yes, 2 = No).

`typmulti` Type of multi-age classroom (1 = 1st-2nd-3rd grades, 2 = 2nd-3rd, 3 = 3rd-4th, 4 = 2nd-3rd- 4th; `NA` when `multi == 2`).

`clenroll` Official class enrollment.

`classize` Project STAR style class size category: 1 = small (roughly 12-17), 2 = regular (roughly 18-22), 3 = regular-larger (roughly 23-26), 4 = large (27 or more). Boundaries follow the STAR classification (Pate-Bain and Achilles, 1986).

`ptratio` Classroom pupil to teacher ratio (IDOE formula: enrollment / [1.00 per full time teacher + 0.33 per full time aide + 0.165 per part time aide]).

`ptia` Prime Time Instructional Aide status: 1 = aide present, 2 = no aide, 3 = other assistant listed. The focal treatment indicator.

`ptstatus` Status of Prime Time aide: 1 = full time in classroom, 2 = part time in classroom; `NA` when `ptia != 1`.

`locale` NCES locale code (1 = large central city, 2 = mid-size central city, 3 = urban fringe of large city, 4 = urban fringe of mid-size city, 5 = large town, 6 = small town, 7 = rural).

`chapter1` School receives Title I (legacy "Chapter 1") money? 1 = Yes, 2 = No.

`ses` School SES, the IDOE percentage of students *not* eligible for subsidized lunch (0 to 100; higher = more affluent). The school level SES variable used in the Lapsley et al. (2002) HLM analyses.

`context` IDOE contextual rank for the school.

`calender` School calendar type (1 = traditional, 2 = year round). Typo preserved from source.

`senroll` Building (school) enrollment.

`sattend` Building attendance rate (percent).

`white1` School percent White.

`black1` School percent Black.

`hispanc1` School percent Hispanic (typo preserved).

`asian1` School percent Asian.

`aindian1` School percent American Indian.

`multi1` School percent multi-racial.

`total1` School total percent non-white.

`noteach` Number of teachers in the building (full time equivalent).

`avgage1` School average teacher age.

`avgexp1` School average teacher experience (years).

`avgsal1` School average teacher salary (dollars).

`spert` School students per teacher.

`thrdclss` Number of third grade classrooms in the building.

`thrdstud` Number of third graders who took ISTEP+ in the building.

`passla1` Building percent passing language arts.

`passmth1` Building percent passing math.

`passbth1` Building percent passing both.

`tmnnce1` Building total battery mean NCE.

`rmdnce1` Building reading *median* NCE. **Note:** the 56 rows from one building (`corp` 5740, `school` 6187) carry the source value 7603, an evident data-entry error in the Indiana DOE file (an NCE is on the 1 to 99 scale, and the same building's other median-NCE columns are in range). The value is preserved as shipped rather than silently corrected, since the true value cannot be recovered; drop or set it to `NA` before analyzing this column.

`lamdnce1` Building language arts *median* NCE.

`mmdnce1` Building mathematics *median* NCE.

`tmdnce1` Building total battery *median* NCE.

`avgcsi1` Building average Cognitive Skills Index.

`geog` Geographic category of the corporation (1 = urban, 2 = suburban, 3 = town, 4 = rural). Sampling stratifier within region.

`totepp` Corporation total expense per pupil (1997–1999, dollars).

`cenroll` Corporation enrollment (all grades).

`cattend` Corporation attendance rate (percent).

`freelnch` Corporation percent eligible for free lunch.

`lep` Corporation percent with limited English proficiency.

`speced` Corporation percent in special education.

`minority` Corporation percent minority.

`white2` Corporation total White public enrollment (raw count).

`black2` Corporation total Black public enrollment.

`hispanc2` Corporation total Hispanic public enrollment (typo preserved).

`asian2` Corporation total Asian public enrollment.

`aindian2` Corporation total American Indian public enrollment.

`multi2` Corporation total multi-racial public enrollment.

`total2` Corporation total non-white public enrollment.

`thrdadm` Corporation third grade ADM (average daily membership).

`thrdtech` Corporation third grade teachers.

`avgage2` Corporation average teacher age.

`avgexp2` Corporation average teacher experience (years).

`avgsal2` Corporation average teacher salary (dollars).

`thrdaide` Corporation third grade aides.

`passla2` Corporation percent passing language arts.

`passmth2` Corporation percent passing math.

`passbth2` Corporation percent passing both.

`tmnnce2` Corporation total battery mean NCE.

`rmdnce2` Corporation reading *median* NCE.

`lamdnce2` Corporation language arts *median* NCE.

`mmdnce2` Corporation mathematics *median* NCE.

`tmdnce2` Corporation total battery *median* NCE.

`rmediate` Corporation remediation funding per pupil (dollars). Typo preserved.

Details

Study background. Indiana's Prime Time program, phased in beginning 1984 to 1985 (Indiana statute; House Bill 1166 of 2001 codified the modern funding formula), was one of the earliest state level initiatives in the United States to use a funding formula to reduce class size and pupil to teacher ratio in Kindergarten through third grade. Funds were distributed to school corporations to maintain a corporation average pupil to teacher ratio of 18:1 in K and grade 1 and 20:1 in grades 2

and 3; corporations could meet the target by hiring additional teachers or, more commonly, paraprofessional instructional assistants. Along with Tennessee's Project STAR (Pate-Bain and Achilles, 1986; covered by Education Week, *Research: Sizing Up Small Classes*, February 2001), Prime Time was a widely cited national model.

In 1999 the Indiana Department of Education funded a three year program evaluation of Prime Time. The third year of the evaluation, conducted by Daniel K. Lapsley and Katrina M. Daytner (Ball State University and Western Illinois University) with technical assistance from Ken Kelley and Scott E. Maxwell (University of Notre Dame), examined the academic performance of randomly selected Indiana third graders on the state mandated ISTEP+ standardized achievement test as a function of class size, pupil to teacher ratio, and the presence of a Prime Time instructional aide, using hierarchical linear modeling. The preliminary report and the AERA 2002 paper that summarizes the analyses are archived as ERIC document ED466679 (Lapsley, Daytner, Kelley, and Maxwell, 2002). An earlier background report from the same evaluation team, prior to the Notre Dame group joining the project, is archived as ERIC document ED455220.

Sampling. School corporations were drawn by stratified cluster sampling with two rules: 25% of corporations from each of the nine Indiana educational service regions, and at least one urban corporation per region, with the remainder proportionally allocated across geographic categories (urban, suburban, town, rural). The achieved sample was 61 corporations (78% of the target), 163 schools, 573 classrooms as counted in the manuscript (586 by the `paste(corp, school, class)` rule used here), and 10,927 students (49.6% female; 85% Caucasian, 9.2% African American, 3.2% Hispanic). 4,016 students were in classrooms with a Prime Time instructional assistant (`ptia == 1`), 6,765 in classrooms without (`ptia == 2`); the file here shows 4,021 and 6,789 plus 117 `ptia == 3` (other assistant listed), with the small differences reflecting cleaning rules applied between the manuscript count and the final SPSS file.

Instruments. Third graders sit for the ISTEP+ (Indiana Statewide Testing for Educational Progress) in September of the school term. The ISTEP+ is published by CTB/McGraw-Hill and includes language arts, reading, and mathematics assessments. Normal Curve Equivalent (NCE) composite scores for these domains and for the total are the `ncread` / `nclang` / `ncmath` / `nctotal` columns and were the criterion variables in the published HLM analyses. NCE scores have a population mean of 50 and a standard deviation of approximately 21.06, with percentiles 1, 50, and 99 mapping to NCE scores of 1, 50, and 99. The `ge*` family is the parallel Gates-MacGinitie battery; the `aa*` family is the African American comparison NCE (AANCE); the `npa*` family is the cognitive abilities battery used as student level covariates in Finch, Bolin, and Kelley (2019).

Nested data structure. The natural hierarchy is student *within* classroom *within* school *within* corporation *within* region. The derived identifiers `corp_id`, `school_id`, and `class_id` are precomputed and safe to use as grouping variables; the bare `corp` and `class` columns are *not* unique by themselves. Class sizes range from 3 to 28 students (median 19); schools have 1 to 8 third grade classrooms (median 3) and 11 to 166 students (median 65); corporations have 15 to 756 students (median 117). Variance decomposition for the published outcome `nctotal` based on the three level random intercept null model `lmer(nctotal ~ 1 + (1 | corp_id/school_id))` gives: between-corporation variance 16.29, between-school within corporation variance 22.72, and within school residual variance 240.43, so that $ICC_{\text{corp}} \approx 0.058$, $ICC_{\text{school|corp}} \approx 0.081$, and the combined cluster $ICC_{\text{cluster}} \approx 0.140$. These nontrivial intraclass correlations are the methodological reason multilevel modeling is preferred to ordinary least squares regression for these data.

Level 1, 2, 3 model framework. For an outcome Y_{ijk} on student i in classroom j in school k, with student level predictor $X^{(1)}_{ijk}$, classroom level predictor $X^{(2)}_{jk}$, and school level predictor $X^{(3)}_{k}$, the

published Lapsley et al. (2002) family of HLM models has the equations

$$Y_{ijk} = \pi_{0jk} + \pi_{1jk} X^{(1)}_{ijk} + e_{ijk} \quad \text{(Level 1)},$$

$$\pi_{0jk} = \beta_{00k} + \beta_{01k} X^{(2)}_{jk} + r_{0jk}, \quad \pi_{1jk} = \beta_{10k} + r_{1jk} \quad \text{(Level 2)},$$

$$\beta_{00k} = \gamma_{000} + \gamma_{001} X^{(3)}_{k} + u_{00k}, \quad \beta_{01k} = \gamma_{010}, \quad \beta_{10k} = \gamma_{100} \quad \text{(Level 3)},$$

with $e_{ijk} \sim N(0, \sigma^2)$, $r_{jk} \sim N(0, \mathbf{T}_\pi)$, and $u_{00k} \sim N(0, \tau_{00})$. Substituting upward, the reduced form is

$$Y_{ijk} = \gamma_{000} + \gamma_{100} X^{(1)}_{ijk} + \gamma_{010} X^{(2)}_{jk} + \gamma_{001} X^{(3)}_{k} + u_{00k} + r_{0jk} + r_{1jk} X^{(1)}_{ijk} + e_{ijk},$$

which in **lme4** translates to `lmer(Y ~ X1 + X2 + X3 + (1 + X1 | corp_id/school_id))`. The examples give concrete fits as commented code, which the help page therefore does not run; uncomment them to fit them.

Suggested benchmark uses. The data set is intentionally rich enough to support a wide range of demonstrations and benchmarks, including:

- Two-, three-, and four-level random intercept and random slope models with **lme4**, **nlme**, or **glmmTMB**.
- Cross-level interaction modeling (e.g., race × class size, ptia × SES).
- ICC, design effect, and cluster level sample size calculations.
- Comparisons of unweighted vs. design weighted estimators for stratified cluster samples.
- Bayesian multilevel modeling and prior sensitivity (**brms**, **MCMCglmm**, **rstanarm**).
- Missing data demonstrations (the `ge*`, `nc*`, and `aa*` columns have non-trivial missingness; see `vapply(prime_time_achievement, function(x) sum(is.na(x)), integer(1))`).
- Multilevel reliability, intraclass correlation, and measurement invariance demonstrations across schools, corporations, and the categorical predictors.

Privacy and identifiability. The student level rows contain no names, addresses, or other personally identifiable information. Demographic variables are age in months, gender, and a six category race code; all other fields are test scores or aggregated school / corporation statistics. The numeric `corp`, `school`, and `class` identifiers are the same administrative numbers used in the original Indiana Department of Education public files for the 2000 to 2001 school year; they could in principle be cross referenced to that public information to identify specific schools or corporations. No individual student can be identified from any combination of variables in this file.

Missing data convention. The Indiana DOE source used `999` as the student level missing data code and `888` as the "not applicable" code for `typmulti` and `ptstatus`. The build script converts both to `NA` (the SPSS missingness ranges already do most of the recoding on import). The retained SPSS variable label is available via `attr(prime_time_achievement$X, "label")` on every variable carried over from the SPSS file (all columns except the derived recode `classize`).

Author(s)

Ken Kelley

Source

Indiana Department of Education program evaluation of Project Prime Time, 2000 to 2001 academic year. Sample of 10,927 third grade students in 586 classrooms in 163 schools in 61 corporations in 9 educational service regions. The records are public data that the author, a member of the evaluation team, is authorized to distribute.

Lapsley, D. K., Daytner, K. M., Kelley, K., and Maxwell, S. E. (2002). *Teacher aides, class size and academic achievement: A preliminary evaluation of Indiana's Prime Time*. Paper presented at the Annual Meeting of the American Educational Research Association, New Orleans, LA, April 1-5, 2002. ERIC document ED466679.

References

Primary citation. Lapsley, D. K., Daytner, K. M., Kelley, K., and Maxwell, S. E. (2002). *Teacher aides, class size and academic achievement: A preliminary evaluation of Indiana's Prime Time*. ERIC document ED466679. https://eric.ed.gov/?id=ED466679.

Use as a multilevel modeling running example. Finch, W. H., Bolin, J. E., and Kelley, K. (2019). *Multilevel modeling using R* (2nd ed.). CRC Press. The 2nd edition (Finch, Bolin, and Kelley, 2019) is the edition that uses these data; later editions are not authored by Kelley and should not be cited for that use.

Background report from the same evaluation team. Lapsley, D. K., and Daytner, K. M. (2001). Indiana's class size reduction initiative: Teacher perspectives on training, implementation, and pedagogy. ERIC document ED455220. https://files.eric.ed.gov/fulltext/ED455220.pdf.

Indiana statutory context. Indiana General Assembly, House Bill 1166 (2001). https://archive.iga.in.gov/2001/bills

Project STAR background and Education Week coverage. Pate-Bain, H., and Achilles, C. M. (1986). Interesting developments on class size. *Phi Delta Kappan, 67*, 662–665. See also Education Week, *Research: Sizing up small classes* (February 7, 2001), https://www.edweek.org/leadership/research-sizing-up-sma

Project STAR teacher aide null result that motivated the Prime Time evaluation. Finn, J. D., Gerber, S. B., Farber, S. L., and Achilles, C. M. (2000). Teacher aides: An alternative to small classes? In M. C. Wang and J. D. Finn (Eds.), *How small classes help teachers do their best* (pp. 131–174). Temple University Center for Research in Human Development and Education.

Examples

```
data(prime_time_achievement)
dim(prime_time_achievement)

# Variable labels from the SPSS source are preserved on every column:
attr(prime_time_achievement$nctotal, "label")
attr(prime_time_achievement$ptia,    "label")

# Cluster counts, reconciled with Lapsley et al., 2002:
length(unique(prime_time_achievement$corp_id))    # 61
length(unique(prime_time_achievement$school_id))  # 163
length(unique(prime_time_achievement$class_id))   # 586

# Reconciling with the manuscript:
table(prime_time_achievement$gender, useNA = "ifany")
table(prime_time_achievement$race,   useNA = "ifany")
```

```
table(prime_time_achievement$ptia)
table(prime_time_achievement$classize)

# ----- Selecting subsets of interest -----

# Caucasian and African American only, the matched race
# supplementary analyses in Lapsley et al., 2002:
pt_wb <- subset(prime_time_achievement, race %in% c(2L, 5L))

# Drop the few "other assistant listed" cases for a clean
# aide / no-aide contrast:
pt_clean <- subset(prime_time_achievement, ptia %in% c(1L, 2L))

# Only rural corporations, coded 4 on geog, which is what the source
# SPSS file's FILTER_$ variable encoded:
pt_rural <- subset(prime_time_achievement, geog == 4L)

# Complete cases on the nctotal-on-race-and-class-size analysis:
analysis_vars <- c("nctotal", "race", "classize", "ses",
                   "corp_id", "school_id", "class_id")
pt_complete <- prime_time_achievement[
  complete.cases(prime_time_achievement[, analysis_vars]),
  analysis_vars
]

# ----- Multilevel fits -----

# Three-level null random intercept model on the full student level
# file. The variance components in the random effects block of the
# summary are the corporation, school within corporation, and residual
# variances behind the intraclass correlations reported in the Details
# section.
m_null <- lme4::lmer(nctotal ~ 1 + (1 | corp_id/school_id),
                     data = prime_time_achievement)
summary(m_null)

# Main effects of race, a student level variable, of ptia and
# classize, classroom level variables, and of ses, a school level
# variable. Compare to Lapsley et al., 2002, which fit closely related
# HLM specifications.
m_main <- lme4::lmer(
  nctotal ~ factor(race) + factor(ptia) + classize + ses +
    (1 | corp_id/school_id),
  data = prime_time_achievement)
summary(m_main)

# Cross-level interaction of ptia with ses. The published finding was
# that the aide benefit was concentrated in higher SES schools, which
# is what the interaction coefficient carries.
m_inter <- lme4::lmer(
  nctotal ~ factor(race) + factor(ptia) * ses + classize +
    (1 | corp_id/school_id),
  data = prime_time_achievement)
```

```
summary(m_inter)
```

`print_anova` *Print a Model Comparison or ANOVA Table With DMAR* p*-value Formatting*

Description

Pretty-print an ANOVA-like object (the output of `stats::anova`, `car::Anova`, `lmerTest::anova`, etc.) with *p*-values formatted at a fixed number of decimal places (default 4) and with a "< 10^(-digits_p)" floor for values too small to express. The default behavior of `print.anova` routes *p*-values through `stats::format.pval`, which applies its own digit rule (`max(1L, getOption("digits") - 2L)`) and switches to scientific notation for tiny values. `print_anova()` sidesteps that by converting the *p*-value columns to character strings up front and printing as a data frame.

Usage

```
print_anova(x, digits_p = 4L)
```

Arguments

`x`	An ANOVA-like data frame with one or more `Pr(...)` columns. Accepts `anova` objects from `stats::anova`, `car::Anova`, `car::Manova`, and `lmerTest::anova`.
`digits_p`	Integer number of decimal places for the *p*-value column(s). Default `4L`.

Details

The returned object is the input `x` invisibly, unchanged: the underlying numeric *p*-values retain full precision and can still be indexed (for example as `x[["Pr(>F)"]]`).

Any column whose name starts with `Pr(` is formatted as a *p*-value column. Other columns print at whatever `getOption("digits")` dictates (so set `options(digits = 4)` for a uniformly compact display).

Value

The input `x`, invisibly and unchanged.

Author(s)

Ken Kelley

See Also

`format_p`, `print_summary`.

Examples

```
fit <- lm(weight ~ Time + Diet, data = ChickWeight)
print_anova(anova(fit))

print_anova(car::Anova(fit, type = "III"))

# Underlying numeric p-values are untouched:
a <- anova(fit)
print_anova(a)
a[["Pr(>F)"]]   # full-precision doubles
```

---

print_summary *Print a Model Summary With DMAR* p*-value Formatting*

Description

Pretty-print a model summary (the output of `summary.lm`, `summary.glm`, or `summary` on an **lme4** or **lmerTest** fit) with *p*-values formatted at a fixed number of decimal places (default 4) and with a "< 10^(-digits_p)" floor for values too small to express. The default `print.summary.lm` / `print.summary.merMod` routes *p*-values through `stats::format.pval`, which applies its own digit rule and switches to scientific notation for tiny values. `print_summary()` sidesteps that by converting the *p*-value columns to character strings up front and printing as a data frame.

Usage

```
print_summary(fit, digits_p = 4L)
```

Arguments

`fit`	A fitted model object with a `summary` method that returns coefficients via `coef(summary(fit))`, including a `Pr(...)` column. Tested with `lm`, `glm`, `lme4::lmer`, and `lmerTest::lmer`.
`digits_p`	Integer number of decimal places for the *p*-value column(s). Default `4L`.

Details

For a linear model, the function prints the coefficient table, the residual standard error and degrees of freedom, the multiple and adjusted R^2, and the omnibus F test and its p-value. For a mixed-effects model fit through **lme4** / **lmerTest**, the function prints the random-effect variances (from `lme4::VarCorr`) and the fixed-effect coefficient table.

The returned object is the model summary, invisibly and unchanged: the underlying numeric *p*-values retain full precision and can still be indexed (for example as `coef(summary(fit))[, "Pr(>|t|)"]`).

Value

The model summary, invisibly and unchanged.

Author(s)

Ken Kelley

See Also

format_p, print_anova.

Examples

```
fit_lm <- lm(weight ~ Time + Diet, data = ChickWeight)
print_summary(fit_lm)

fit_lmer <- lme4::lmer(weight ~ Time + (1 | Chick), data = ChickWeight)
print_summary(fit_lmer)

# Underlying numeric p-values are untouched:
sm <- summary(fit_lm)
sm$coefficients[, "Pr(>|t|)"]   # full-precision doubles
```

---

probability_of_superiority_paired

Probability of Superiority for a Paired-Samples Design

Description

Computes the probability-of-superiority effect size for paired observations (Grissom & Kim, 2005, 2012), $P_S = \Pr(Y_1 > Y_2)$, along with an analytic confidence interval based on the Brunner-Munzel (2000) U-statistic standard error and a Fisher- $\operatorname{arctanh}$ transformation to keep the bounds inside $[0, 1]$. The paired counterpart of the Vargha-Delaney (2000) A statistic / cliff_delta for two independent groups.

Usage

```
probability_of_superiority_paired(x, y, conf_level = 0.95)
```

Arguments

x, y	Paired numeric vectors of equal length. x and y are interpreted as repeated measurements on the same units (e.g., pre/post, condition 1 / condition 2, sibling-1 / sibling-2).
conf_level	Confidence level. Default 0.95.

Details

Definition. For paired observations (x_i, y_i),

$$P_S \;=\; \Pr(Y > X) + 0.5 \cdot \Pr(Y = X),$$

where ties are split. The sample estimator is the proportion of pairs with $y_i > x_i$, plus half the proportion of ties. This is the natural paired-data analog of Vargha-Delaney's A statistic and is unbiased under exchangeability of paired observations.

Why paired-specific. The independent-groups `cles` and `cliff_delta` estimators are biased when the two samples are paired, because their variance formulas assume independence of the two groups. For paired data the within-pair correlation reduces the effective sampling variance, which is captured by the Brunner-Munzel (2000) variance used here.

Confidence interval. The standard error is built from the within-pair sign indicators (Brunner-Munzel, 2000):

$$\mathrm{Var}(\hat{P}_S) \;=\; \frac{1}{n^2}\sum_{i=1}^{n}(s_i - \bar{s})^2,$$

where $s_i = \mathrm{I}(y_i > x_i) + 0.5 \cdot \mathrm{I}(y_i = x_i)$. The CI is built on the $\operatorname{arctanh}(2P_S - 1)$ scale (mapping $P_S \in [0, 1]$ to the real line) and back-transformed to keep the limits inside the unit interval, exactly mirroring `cliff_delta`.

Value

A `data.frame` with rows for the point estimate of P_S, the lower / upper CI bounds, the variance, and the counts of within-pair wins / ties / losses for y_1.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Brunner, E., & Munzel, U. (2000). The nonparametric Behrens-Fisher problem: Asymptotic theory and a small-sample approximation. *Biometrical Journal, 42*(1), 17–25. doi:10.1002/(SICI)1521-4036(200001)42:1<17::AIDBIMJ17>3.0.CO;2U

Grissom, R. J., & Kim, J. J. (2005). *Effect sizes for research: A broad practical approach*. Lawrence Erlbaum.

Grissom, R. J., & Kim, J. J. (2012). *Effect sizes for research: Univariate and multivariate applications* (2nd ed.). Routledge.

Vargha, A., & Delaney, H. D. (2000). A critique and improvement of the CL common language effect size statistics of McGraw and Wong. *Journal of Educational and Behavioral Statistics, 25*(2), 101–132. doi:10.3102/10769986025002101

See Also

`cliff_delta`, `cles`, `proportion_of_superiority`

Other effect size estimates: `cles()`, `cliff_delta()`, `correction_for_attenuation()`, `eta_squared()`, `eta_squared_generalized()`, `eta_squared_partial()`, `expected_partial_r()`, `expected_r()`,

`expected_smd()`, `nnt_from_smd()`, `omega_squared()`, `omega_squared_partial()`, `proportion_of_superiority()`, `responder_analysis()`, `smd_trimmed()`

Examples

```
# 1. Paired pre/post data:
set.seed(113)
pre  <- rnorm(30, mean = 100, sd = 15)
post <- pre + rnorm(30, mean =  5, sd = 10)
probability_of_superiority_paired(x = pre, y = post)
```

`procrustes_phi` *Tucker's Congruence Coefficient ϕ (Factor Similarity)*

Description

Computes Tucker's (1951) congruence coefficient ϕ, a measure of similarity between two factor-loading patterns (typically the standardized loadings of the same factor estimated on two different samples or with different methods), together with a permutation-based *p*-value testing the null hypothesis of unrelated loading patterns. ϕ is the standard tool for factor-replication studies (Lorenzo-Seva & ten Berge, 2006).

Usage

```
procrustes_phi(loadings_1, loadings_2, n_perm = 10000L)
```

Arguments

`loadings_1`, `loadings_2`
: Numeric vectors of factor loadings on the same indicator set, of equal length. Either standardized or raw loadings work; the coefficient is scale-invariant.

`n_perm`
: Number of permutations for the significance test. Default `10000`. Set to 0 to skip the test.

Details

Definition. For two vectors of loadings $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$ on a shared set of p indicators, Tucker's congruence coefficient is

$$\phi(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) \;=\; \frac{\sum_{i=1}^{p} \lambda_{1i}\lambda_{2i}}{\sqrt{\sum_{i=1}^{p} \lambda_{1i}^2 \cdot \sum_{i=1}^{p} \lambda_{2i}^2}}.$$

ϕ is the cosine of the angle between the two loading vectors and ranges over $[-1, 1]$; values near ± 1 indicate high (anti-)congruence, values near 0 indicate orthogonality.

Permutation test. Under the null hypothesis that the two loading patterns are unrelated, randomly permuting one of the loading vectors and recomputing ϕ produces a sampling distribution against which the observed ϕ can be evaluated. The two-sided *p*-value is $(r + 1)/(m + 1)$, where *r* counts the permuted $|\phi|$ values at least as large as the observed $|\phi|$ and *m* is `n_perm`. Adding one to each

part counts the observed arrangement, which is itself a legitimate permutation; without it a p-value of exactly zero could be reported, a value a sampled permutation test cannot support (Phipson & Smyth, 2010). The smallest reportable p-value is therefore $1/(m+1)$.

Interpretation. Benchmark values for ϕ have been proposed in the literature (Lorenzo-Seva & ten Berge, 2006), but context always matters; this package reports the coefficient with its uncertainty and leaves interpretation to the context of the application.

Value

A `data.frame` (class `dmar_tbl`) in `term` / `value` layout with the row `tucker_phi`, the point estimate of ϕ. When `n_perm > 0` the table also carries `p_value_perm`, the two-sided permutation p-value, and `n_perm`, the number of permutations requested.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Lorenzo-Seva, U., & ten Berge, J. M. F. (2006). Tucker's congruence coefficient as a meaningful index of factor similarity. *Methodology, 2*(2), 57–64. doi:10.1027/16142241.2.2.57

Phipson, B., & Smyth, G. K. (2010). Permutation p-values should never be zero: Calculating exact p-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology, 9*(1), Article 39. doi:10.2202/15446115.1585

Tucker, L. R. (1951). *A method for synthesis of factor analysis studies* (Personnel Research Section Report No. 984). Department of the Army.

See Also

`cor`, `reliability_H`

Other multivariate and latent variable methods: `average_variance_extracted()`, `bifactor_indices()`, `cfa_1()`, `cfa_2()`, `cfa_k()`, `ci_eigenvalue()`, `common_method_marker()`, `common_method_single_factor()`, `dmacs()`, `ecvi()`, `htmt()`, `irt_grm()`, `irt_information()`, `measurement_alignment()`, `measurement_invariance()`, `simple_structure()`

Examples

```
set.seed(113)
# 1. Two highly similar loading patterns:
l1 <- c(0.72, 0.65, 0.81, 0.55, 0.69)
l2 <- c(0.70, 0.62, 0.83, 0.58, 0.66)
procrustes_phi(l1, l2)

# 2. Loadings on different factors should show low congruence:
l3 <- c(0.10, 0.05, 0.20, 0.85, 0.78)
procrustes_phi(l1, l3, n_perm = 5000)
```

`proportion_of_superiority` *Proportion of Superiority (Sometimes Called Cohen's U_3)*

Description

Computes the proportion of the treatment-group population that exceeds the *control-group mean* under bivariate normality with equal variances. This quantity is sometimes called Cohen's U_3 (Cohen, 1988). Under those assumptions it equals $\Phi(\delta)$, where δ is the population standardized mean difference. When sample sizes are supplied, the CI on the proportion of superiority is constructed by transforming the noncentral *t* CI on Cohen's *d* via Φ, which is monotone and therefore preserves coverage exactly.

Usage

```
proportion_of_superiority(
  smd,
  n_1 = NULL,
  n_2 = NULL,
  conf_level = 0.95,
  smd_lower = NULL,
  smd_upper = NULL
)
```

Arguments

`smd`	Sample standardized mean difference (Cohen's *d*). Numeric scalar.
`n_1`, `n_2`	Group sample sizes (required if a CI is wanted).
`conf_level`	Confidence level for the CI. Default `0.95`.
`smd_lower`, `smd_upper`	Optional pre-computed CI limits on *d*; when supplied directly, the function skips the noncentral *t* step and just transforms these limits.

Details

The proportion of superiority is one of three "U" indices Cohen (1988) defined; the other two (U_1, the proportion of non-overlap, and U_2, the proportion of either population that exceeds the same percentile in the other) can be derived from it directly: with $U_3 = \Phi(\delta)$ for the proportion of superiority, Cohen's $U_2 = \Phi(\delta/2)$ and $U_1 = (2 \cdot U_2 - 1)/U_2$ (Cohen, 1988, Table 2.2.1).

Why this rather than `cles`. The proportion of superiority answers the question "what fraction of the treatment population exceeds the *control-group mean*," whereas `cles` answers "what fraction of randomly drawn pairs favor the treatment over the control." Both are unitless probability-scale summaries of a Cohen's-*d* difference, but the proportion of superiority is marginal while CLES is paired. Specifically, $\Phi(d)$ versus $\Phi(d/\sqrt{2})$; for $d = 0.5$, the proportion of superiority is 0.69 and CLES is 0.64.

CI construction. Because $\Phi(\cdot)$ is monotone, the CI on the proportion of superiority is just $[\Phi(d_L),\ \Phi(d_U)]$ where $[d_L,\ d_U]$ is the noncentral *t* CI on *d* from `ci_smd`.

Value

A `data.frame` with rows for *d*, the proportion of superiority, and (when a CI is constructable) the lower / upper limits on *d* and on the proportion of superiority.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum. (See Section 2.2 for the U_1, U_2, and U_3 indices.)

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. Academic Press.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. (The noncentral *t* interval on the standardized mean difference that is transformed here.) doi:10.18637/jss.v020.i08

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

`cles`, `nnt_from_smd`, `smd`, `ci_smd`, `probability_of_superiority_paired`

Other effect size estimates: `cles()`, `cliff_delta()`, `correction_for_attenuation()`, `eta_squared()`, `eta_squared_generalized()`, `eta_squared_partial()`, `expected_partial_r()`, `expected_r()`, `expected_smd()`, `nnt_from_smd()`, `omega_squared()`, `omega_squared_partial()`, `probability_of_superiority_pai`
`responder_analysis()`, `smd_trimmed()`

Examples

```
# 1. Proportion of superiority at three reference d values:
proportion_of_superiority(smd = 0.2)
proportion_of_superiority(smd = 0.5)
proportion_of_superiority(smd = 0.8)

# 2. With a noncentral t CI from sample sizes:
proportion_of_superiority(smd = 0.5, n_1 = 50, n_2 = 50, conf_level = 0.95)
```

`pygmalion` *Pygmalion in the Classroom Teacher-Expectancy Data*

Description

The teacher-expectancy data from Rosenthal and Jacobson's (1968) *Pygmalion in the Classroom*, the study that introduced the "Pygmalion effect": the hypothesis that a teacher's expectations can become a self-fulfilling prophecy for a pupil's intellectual growth. Intelligence-test scores were obtained for $N = 310$ elementary school children in grades 1 through 6, of whom $n = 64$ were randomly designated to their teachers as likely "intellectual bloomers" while the remaining $n = 246$ served as controls. The data set is a classic benchmark for the analysis of covariance (ANCOVA) and, in particular, for ANCOVA with *heterogeneity of regression*: it is the running example for that topic in Maxwell, Delaney, and Kelley, *Designing Experiments and Analyzing Data: A Model Comparison Perspective* (Routledge), where it appears as a Chapter 9 example (and as a Chapter 3 exercise).

Usage

```
pygmalion
```

Format

A data frame with 310 observations on 6 variables.

`grade` Grade in school at the start of the study, an integer from 1 to 6.

`treatment` Factor with levels `Control` (reference, $n = 246$) and `Bloomer` ($n = 64$). The `Bloomer` children were a randomly selected ~20% of each classroom whose teachers were told, on the basis of a fictitious test purportedly predicting intellectual blooming, that they were likely to show unusual gains during the year; the `Control` children were not singled out. In the AMCP source this variable is coded `1` = Bloomer, `0` = Control.

`iq_pre` Pretest total IQ, measured before the expectancy manipulation. The covariate in the analysis of covariance.

`iq_4` Total IQ at an intermediate follow-up assessment.

`iq_8` Total IQ at the end-of-study follow-up assessment. This is the dependent variable in the book's Chapter 9 analysis of covariance.

`iq_gain` Total IQ change from pretest to the end-of-study assessment, equal to `iq_8 - iq_pre`.

Details

The study. Robert Rosenthal (Harvard University) and Lenore Jacobson (principal of an elementary school in South San Francisco referred to as "Oak School") set out to test experimentally whether teacher expectations influence pupil achievement. At the start of the school year all children were given a standardized test of general ability, described to teachers as the "Harvard Test of Inflected Acquisition," a test said to identify children poised for an intellectual growth spurt. In reality the instrument was Flanagan's Tests of General Ability (TOGA) and the children identified as likely "bloomers" were chosen *at random*, about one in five per classroom. The only experimental manipulation was the expectation planted in the teachers' minds. Children were re-tested over the following year(s), and the question was whether the randomly labeled bloomers would out-gain their controls in measured IQ. Rosenthal and Jacobson reported that they did, most strongly in the earliest grades, and interpreted the difference as evidence that teacher expectations operate as a self-fulfilling prophecy. The study became one of the most famous and most debated experiments

in the social sciences; subsequent critiques (e.g., Thorndike, 1968) questioned the reliability of the TOGA at the extremes of the score range for the youngest children, which is itself part of why the data are instructive for teaching careful analysis.

Why it is a benchmark for heterogeneity of regression. A standard ANCOVA adjusts the group comparison for the pretest covariate under the assumption that the regression of the outcome on the covariate has the *same* slope in every group (homogeneity of regression). In these data that assumption is questionable: the within-group regression of `iq_8` on `iq_pre` is steeper for the bloomers than for the controls, so the estimated treatment effect depends on the covariate value at which it is evaluated. This makes the data an ideal teaching example for (a) testing the homogeneity-of-regression assumption, (b) interpreting a treatment-by-covariate interaction, and (c) estimating the treatment effect, and its sampling variance, *at chosen covariate values* rather than only at the grand mean.

Reproducible quantities. Fitting the separate-slopes model `lm(iq_8 ~ iq_pre * treatment)` gives a within-group slope of 0.77799 for the controls and 0.96894 for the bloomers (reported as 0.96895 in `MBESS::var.ete`, a fifth-decimal rounding difference). The pooled within-group residual variance is $\hat{\sigma}^2 = 175.3251$ on 306 degrees of freedom, and the sample variance of the covariate is 348.91. These are exactly the inputs used in the worked example for the variance of the estimated treatment effect at selected covariate values under heterogeneity of regression (Li, McLouth, and Delaney; see `MBESS::var.ete`).

Relationship to the AMCP package. The same numeric data ship with the book's data companion, the **AMCP** package, as `chapter_9_exercise_15` and `chapter_9_extension_exercise_3` (with `IQGain`) and `chapter_3_exercise_22` (without it). The version here renames the columns to DMAR's descriptive snake_case style and labels the experimental condition as a factor; no measured value has been altered. See `data-raw/pygmalion.R` for the construction script and its verification checks.

Author(s)

Ken Kelley

Source

Rosenthal, R., & Jacobson, L. (1968). *Pygmalion in the classroom: Teacher expectation and pupils' intellectual development*. Holt, Rinehart and Winston.

Distributed with the **AMCP** data companion to Maxwell, Delaney, and Kelley (see References) as `chapter_9_exercise_15`.

References

Rosenthal, R., & Jacobson, L. (1968). *Pygmalion in the classroom: Teacher expectation and pupils' intellectual development*. Holt, Rinehart and Winston.

Rosenthal, R., & Jacobson, L. (1968). Pygmalion in the classroom. *The Urban Review, 3*(1), 16–20. doi:10.1007/BF02322211

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (Heterogeneity-of-regression ANCOVA example, Chapter 9.)

Thorndike, R. L. (1968). Review of *Pygmalion in the Classroom. American Educational Research Journal, 5*(4), 708–711.

See Also

ancova for an ANCOVA that returns adjusted means, effect size confidence intervals, and a homogeneity-of-regression test.

Examples

```
data(pygmalion)
str(pygmalion)

# Design: pupils per condition within each grade.
table(pygmalion$treatment, pygmalion$grade)

# ---- Heterogeneity-of-regression ANCOVA (book Chapter 9) ----
# Separate IQ8-on-IQpre slopes for the two conditions.
fit_het <- lm(iq_8 ~ iq_pre * treatment, data = pygmalion)
coef(fit_het)
# Control slope = 0.778; the interaction (0.191) gives the
# steeper Bloomer slope of 0.969.

# The treatment-by-covariate interaction is the
# heterogeneity-of-regression test (1 df): compare the additive
# ANCOVA model to the separate-slopes model.
fit_add <- lm(iq_8 ~ iq_pre + treatment, data = pygmalion)
anova(fit_add, fit_het)

# Pooled within-group residual variance (175.3251) and the
# covariate variance (348.91), as used by MBESS::var.ete.
sum(residuals(fit_het)^2) / fit_het$df.residual
var(pygmalion$iq_pre)

# ---- DMAR's ANCOVA, with the homogeneity-of-regression check ----
ancova(pygmalion, outcome = "iq_8", treatment = "treatment",
       covariates = "iq_pre")
```

R2_mixed_effects *Marginal and Conditional R^2 for a Mixed-Effects Model*

Description

Computes the Nakagawa and Schielzeth (2013) marginal and conditional coefficients of determination from a fitted mixed-effects model, returned in tidy long form. The marginal R^2 is the proportion of total variance explained by the fixed effects alone; the conditional R^2 is the proportion explained by the fixed and random effects together. The two quantities are the mixed-effects companions to the intraclass correlation returned by icc_lmer: where the ICC isolates the share of variance attributable to a single grouping factor, the marginal and conditional R^2 summarize how much of the outcome variance the fixed and random parts of the model account for.

Usage

```
R2_mixed_effects(
  model,
  conf_level = 0.95,
  ci_method = c("none", "boot"),
  B = 1000,
  seed = NULL,
  ...
)
```

Arguments

model	A fitted mixed-effects model. A lmer fit (class lmerMod) is the primary target; an lme fit is also accepted.
conf_level	Confidence level. Default 0.95. Used only when a bootstrap interval is requested (see ci_method).
ci_method	Interval method for the two R^2 values. The default "none" returns point estimates only. "boot" adds a parametric bootstrap percentile interval (lmerMod models only, via bootMer); see *Details*.
B	Number of bootstrap replications when ci_method = "boot". Default 1000.
seed	Optional integer seed for the bootstrap. Default NULL, meaning the caller's current RNG state is used and left unchanged. When supplied, the seed is set inside the function and the caller's RNG state is restored on exit.
...	Currently unused.

Details

Variance decomposition. Writing σ_f^2 for the variance of the fixed-effect linear predictor, σ_r^2 for the variance attributable to the random effects, and σ_ε^2 for the residual variance,

$$R^2_{\text{marginal}} = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_r^2 + \sigma_\varepsilon^2}, \qquad R^2_{\text{conditional}} = \frac{\sigma_f^2 + \sigma_r^2}{\sigma_f^2 + \sigma_r^2 + \sigma_\varepsilon^2}.$$

The fixed-effect variance is $\sigma_f^2 = \text{var}(\mathbf{X}\boldsymbol{\beta})$, the variance of the fitted fixed-effect linear predictor across the observations. The residual variance is $\sigma_\varepsilon^2 = \text{sigma}(\text{model})^2$.

Random-effect variance. For a random-intercept model the random-effect variance is the sum of the variance components read off VarCorr. For a model with random slopes the variance contributed by a random-effects term depends on the values of the associated covariates, so the sum of the diagonal variance components is not correct on its own. This function uses the Johnson (2014) extension: for each random-effects term with design matrix $\mathbf{Z}$ and estimated covariance matrix $\boldsymbol{\Sigma}$, its contribution is the mean over the observations of the quadratic form $\mathbf{z}_i^\top \boldsymbol{\Sigma}\, \mathbf{z}_i$, that is, $\frac{1}{n}\,\text{tr}(\mathbf{Z}\boldsymbol{\Sigma}\mathbf{Z}^\top)$, and σ_r^2 is the sum of these contributions across all random-effects terms. For a random-intercept term this reduces to the intercept variance component, so the two paths agree.

Scope. The decomposition here is the one appropriate for a Gaussian (identity-link) linear mixed model, which is what lmer and lme fit. Generalized linear mixed models introduce a distribution-specific variance term and are not handled by this function.

The bootstrap interval. The default `ci_method = "none"` reports the two point estimates alone, so the bootstrap is what to ask for when the marginal and conditional R^2 are to be reported with an interval and the refits it costs are affordable. With `ci_method = "boot"` the interval comes from a parametric bootstrap (`bootMer`): each of the `B` replicates (1000 by default) simulates a new response vector from the fitted model, refits the model, and recomputes the two R^2 values. The unit of resampling is therefore a whole simulated data set drawn from the estimated model, not a resampled set of cases. Only the percentile interval is offered: the limits are the empirical quantiles of the `B` bootstrap values (Efron & Tibshirani, 1993); there is no BCa or bootstrap standard error variant. Replicates whose refit fails are dropped, and the interval is computed from the replications that return a value. The default `B = 1000` is adequate for the central quantiles a percentile interval uses; raising it tightens the Monte Carlo error of the reported limits. Bootstrap results vary from run to run; supply `seed` for reproducibility.

Value

A `data.frame` with rows `"R2_marginal"` and `"R2_conditional"` in the `value` column. When `ci_method = "boot"`, lower- and upper-limit rows for each quantity are appended and the confidence level is carried on the object.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, NY: Chapman & Hall/CRC.

Johnson, P. C. D. (2014). Extension of Nakagawa & Schielzeth's R^2_{GLMM} to random slopes models. *Methods in Ecology and Evolution, 5*(9), 944–946. doi:10.1111/2041210X.12225

Nakagawa, S., & Schielzeth, H. (2013). A general and simple method for obtaining R^2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution, 4*(2), 133–142. doi:10.1111/j.2041210x.2012.00261.x

See Also

`icc_lmer`, `ss_power_mixed_effects`, `lmer`, `VarCorr`

Other agreement and measurement: `content_validity_index()`, `gwet_ac()`, `icc_lmer()`, `krippendorff_alpha()`, `limits_of_agreement()`, `lin_ccc()`, `variance_components_mls()`

Other mixed models: `R2_mixed_effects_decomposition()`, `icc_lmer()`, `manova_split_plot()`, `mixed_anova()`, `ss_aipe_mixed_effects()`, `ss_aipe_mixed_effects_sensitivity()`, `ss_power_mixed_effects()`, `ss_power_split_plot_anova()`

Examples

```
fit <- lme4::lmer(Reaction ~ Days + (Days | Subject),
                  data = lme4::sleepstudy)

# Marginal R2 is the proportion of variance the fixed effects account for;
# conditional R2 adds what the random effects account for, so the gap
```

```
# between the two is what the subject-level terms buy.
R2_mixed_effects(fit)

# A parametric bootstrap percentile interval for both quantities. Each
# replication refits the model, so B = 20 keeps the example quick; a
# reported interval deserves the default B = 1000, and raising B
# further tightens the Monte Carlo error of the limits. The seed makes
# the limits reproducible and leaves the caller's generator state as
# it was.
R2_mixed_effects(fit, ci_method = "boot", B = 20, seed = 113)
```

`R2_mixed_effects_decomposition` *R-Squared Measures for Mixed-Effects Models*

Description

Computes the Rights and Sterba (2019) integrative framework of R-squared measures for a fitted two-level linear mixed-effects (multilevel) model. The model implied outcome variance is fully decomposed into five sources: variance due to level-1 predictors via fixed slopes (f_1), level-2 predictors via fixed slopes (f_2), predictors via random slope (co)variation (v), cluster-specific outcome means via random intercept variation (m), and level-1 residuals (σ^2). Proportions of the total, within-cluster, and between-cluster outcome variance attributable to combinations of these sources give the family of R-squared measures.

Usage

```
R2_mixed_effects_decomposition(model)
```

Arguments

`model`	A fitted two-level model of class `merMod` (from **lme4**, e.g.\ `lmer`) or `lme` (from **nlme**). The model must use numeric predictors (only the cluster variable may be a factor) and must not contain `I()` terms; create any transformed predictors as their own columns first.

Details

The companion `R2_mixed_effects` returns the Nakagawa and Schielzeth marginal and conditional R-squared; this function contains those two as special cases (`total_f` and `total_fvm`) within the fuller source decomposition.

The measure superscripts index the variance sources in the numerator and the subscripts index the outcome variance in the denominator: `total_*` use the total outcome variance, `within_*` the within-cluster variance ($f_1 + v + \sigma^2$), and `between_*` the between-cluster variance ($f_2 + m$). `total_fvm` is the omnibus measure (all explained sources over the total variance) and, for a random-intercept model, coincides with the Nakagawa and Schielzeth conditional R-squared computed by

R2_mixed_effects; total_f coincides with their marginal R-squared. See Rights and Sterba (2019, Table 1) for the definitions.

The measures were derived under the assumption that the fitted model uses cluster-mean-centering of the level-1 predictors (with the cluster means entered as level-2 predictors). When that centering is not detected, only the total-variance measures are returned, matching the reference implementation.

Fitting the model requires **lme4** (for a merMod fit) or **nlme** (for an lme fit) to be installed.

Value

A data.frame (dmar_tbl) with columns term and value. When the level-1 predictors are cluster-mean-centered, the full set of 12 measures is returned, named total_f1, total_f2, total_v, total_m, total_f, total_fv, total_fvm, within_f1, within_v, within_fv, between_f2, and between_m; otherwise the five total-variance measures total_f, total_v, total_m, total_fv, and total_fvm are returned (the within/between split requires cluster-mean-centering). The returned object carries the source-by-target variance decomposition in attr(x, "decomposition").

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Rights, J. D., & Sterba, S. K. (2019). Quantifying explained variance in multilevel models: An integrative framework for defining R-squared measures. *Psychological Methods, 24*(3), 309–338. doi:10.1037/met0000184

Nakagawa, S., & Schielzeth, H. (2013). A general and simple method for obtaining R^2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution, 4*(2), 133–142. doi:10.1111/j.2041210x.2012.00261.x

See Also

R2_mixed_effects, icc_lmer

Other mixed models: R2_mixed_effects(), icc_lmer(), manova_split_plot(), mixed_anova(), ss_aipe_mixed_effects(), ss_aipe_mixed_effects_sensitivity(), ss_power_mixed_effects(), ss_power_split_plot_anova()

Examples

```
fit <- lme4::lmer(Reaction ~ Days + (Days | Subject),
                  data = lme4::sleepstudy)
R2_mixed_effects_decomposition(fit)
```

`randomization_test` *Randomization (Permutation) Test for Two Independent Groups*

Description

Compares two independent groups by referring the observed statistic to the distribution of that same statistic over reassignments of the observed scores to the two groups. That reference distribution, not a normal or a *t* distribution, supplies the *p*-value, so the test needs no assumption about the shape of the population. Alongside the test, the function reports the effect sizes that answer the question the test only screens: the mean difference and its randomization-based interval, the standardized mean difference with a noncentral *t* interval, the common language effect size, and Cliff's delta.

Usage

```
randomization_test(
  x = NULL,
  group = NULL,
  data = NULL,
  group_1 = NULL,
  group_2 = NULL,
  statistic = c("mean", "t"),
  alternative = c("two_sided", "less", "greater"),
  exact = NULL,
  n_resamples = 10000L,
  seed = NULL,
  conf_level = 0.95,
  shift_ci = TRUE
)
```

Arguments

`x`	A formula of the form `y ~ group`, or a numeric response vector to be paired with `group`. Leave `NULL` when the two samples are supplied through `group_1` and `group_2`.
`group`	A grouping variable the same length as `x`, with exactly two levels after unused levels are dropped. Ignored when `x` is a formula.
`data`	An optional `data.frame` in which to find the variables named in the formula.
`group_1, group_2`	The two samples supplied directly as numeric vectors, an alternative to the formula and response-plus-grouping interfaces. Lengths need not be equal.
`statistic`	One of `"mean"` (default) or `"t"`. `"mean"` uses the difference in group means. `"t"` uses the studentized (Welch) statistic, which divides that difference by its separate-variances standard error and is the better choice when the groups may differ in variance (see Details).

`alternative`	One of `"two_sided"` (default; the base-R spelling `"two.sided"` is accepted as an alias), `"less"`, or `"greater"`. The direction refers to the first group minus the second, the same orientation `t.test` uses.
`exact`	Logical. If `NULL` (default), the test enumerates every reassignment when `choose(N, n_1)` is at most 50,000 and samples reassignments otherwise. If `TRUE`, enumeration is forced (refused above 1,000,000 reassignments). If `FALSE`, Monte Carlo is forced.
`n_resamples`	Number of randomly drawn reassignments when enumeration is not used. Default `10000L`.
`seed`	Optional integer seed for the Monte Carlo branch. Default `NULL`, which leaves the user's current RNG state intact; supply an integer for reproducibility. When a seed is supplied the RNG state in place before the call is restored on exit.
`conf_level`	Confidence level for every interval reported, the inverted randomization interval included. Default `0.95`.
`shift_ci`	Logical. Compute the randomization-based interval for the shift by inverting the test? Default `TRUE`. Setting it to `FALSE` reports the two endpoints as `NA` and skips the inversion, which is the expensive part of the call.

Details

What the randomization distribution is. Suppose *N* participants were randomly assigned, n_1 to one condition and n_2 to the other. Under the null hypothesis that the condition a participant received made no difference to that participant's score, each score would have been the same number no matter which group the participant landed in. The assignment actually used was one draw from the $\binom{N}{n_1}$ assignments the randomization could equally well have produced, so every one of those assignments was equally likely, and each of them yields a value of the test statistic. Those values are the randomization distribution. The *p*-value is the proportion of them at least as extreme as the value the experiment actually produced.

Why there is no normality assumption. Nothing in that argument mentions a population, a normal curve, or a sampling model. The probability comes from the coin flips the experimenter performed, which are known exactly because the experimenter performed them. This is the inferential logic Fisher (1935) used to introduce experimental design, and it is where Chapter 1 of Maxwell, Delaney, and Kelley (2027) starts, for the same reason: the validity of the test rests on the randomization rather than on assumptions a data analyst cannot check.

What the test does and does not license. A small *p*-value says the observed separation between the groups would rarely arise from reassignment alone, which is evidence that the assignment mattered. It does not say how much it mattered, and with a large *N* an uninteresting difference will produce a small *p*-value. It also does not, by itself, license generalization beyond the participants at hand: randomization licenses a causal claim about these units, while generalization to a population is a separate argument that rests on how the units were recruited. That is why this function reports effect sizes with intervals rather than a *p*-value alone.

Why the studentized statistic. With $n_1 = n_2$ and equal population variances the two statistics give the same *p*-value to within the discreteness of the reference distribution, because the denominator of the studentized statistic is then nearly constant across reassignments. When the variances differ and the groups are unbalanced they part company. Reassigning scores between groups of unequal size mixes the two variances in proportions that the observed assignment does not have, so the reference

distribution for the raw mean difference is built under a null that is false in a second way, and the test's actual Type I error rate drifts away from the nominal level. The studentized statistic rescales each reassignment by its own separate-variances standard error, which removes most of that drift and remains asymptotically valid under heteroscedasticity (Janssen, 1997; Neuhaus, 1993). Use `statistic = "t"` whenever unequal variances are plausible, which for unbalanced designs is nearly always.

Exact or Monte Carlo. When `choose(N, n_1)` is at most 50,000 every reassignment is enumerated and the *p*-value is exact: it is a count divided by a known total, with no approximation anywhere. Above that threshold `n_resamples` reassignments are drawn at random and the *p*-value is $(r + 1)/(m + 1)$, where *r* counts the sampled reassignments at least as extreme as the observed one and *m* is `n_resamples`. Adding one to each part counts the observed assignment, which is itself a legitimate reassignment; without it a *p*-value of exactly zero could be reported for a hypothesis the data cannot rule out, and the test would be anticonservative (Phipson & Smyth, 2010). The reported `p_value_se` is $\sqrt{\hat{p}(1-\hat{p})/m}$, the standard error of the resampling itself. It describes how much the *p*-value would move if the reassignments were drawn again, not how much it would move in a new experiment. Raising `n_resamples` shrinks it at the usual $1/\sqrt{m}$ rate.

The randomization interval, and how it differs from the normal theory one. Suppose the treatment adds a constant δ to every score it touches. Subtracting δ from each first-group score should then leave scores that are exchangeable across groups, so the randomization test applied to the subtracted data is a test of $H_0 : \text{shift} = \delta$. The set of δ for which that test does not reject at level $1-$ `conf_level` is a confidence interval for the shift, and it is reported as `shift_lower_limit` and `shift_upper_limit`. Inverting a test this way is the general recipe (Ernst, 2004); the endpoints are located by bisection on the *p*-value, using the same reassignments throughout so the interval and the test agree.

The contrast with `normal_theory_lower_limit` and `normal_theory_upper_limit`, which are Welch's *t* limits on the same mean difference, is worth reading whenever both are printed. The randomization interval is exactly the set of shifts the test being run does not reject, so the test and the interval can never disagree. The normal theory interval instead assumes the sampling distribution of the mean difference has a known shape; it is smooth, symmetric about the point estimate, and can extend past the range the data can support. The randomization interval is discrete, need not be symmetric, and in a very small design is unbounded, a correct statement of how little information the design carries rather than a defect: with three observations per group the smallest attainable two-sided *p*-value is 2/20 = 0.10, so no shift can be rejected at the 5% level and the 95% interval is the whole real line. The randomization interval also inherits the shift model, so it answers a narrower question than the test does: the test needs only exchangeability, while the interval needs the treatment to move every score by the same amount.

Effect sizes. Every effect size reported here comes from the package function that owns it, so the numbers match a direct call. `smd` and `ci_smd` supply the standardized mean difference and its noncentral *t* interval; `cles` supplies the common language effect size, the probability that a randomly drawn score from the first group exceeds one from the second, by transforming those limits through $\Phi(\cdot/\sqrt{2})$; and `cliff_delta` supplies Cliff's delta with its consistent interval. The standardized mean difference and the common language effect size are normal theory quantities, so their intervals lean on the assumption the test itself avoids. Cliff's delta does not: it is a function of the ordering of the observations alone, which makes it the natural effect size companion to a randomization test. Reporting all three lets a reader see whether the distribution-free and normal theory summaries tell the same story.

Ranks give the Wilcoxon test. Replacing the scores by their ranks and running this test with

`statistic = "mean"` reproduces the exact Wilcoxon rank sum test, since the rank sum is a monotone function of the difference in mean ranks. That equivalence is a useful check and a reminder of what the rank test is: a randomization test on transformed data.

Value

A `data.frame` with a `term` column and a numeric `value` column, in three blocks.

The test: `mean_difference` (first group minus second), `statistic` (the statistic actually referred to the reference distribution), `p_value`, and `p_value_se` (the Monte Carlo standard error of the *p*-value, `NA` under exact enumeration, which has no Monte Carlo error).

The intervals and effect sizes: `shift_lower_limit` and `shift_upper_limit` (the randomization interval for the shift, obtained by inverting the test); `normal_theory_lower_limit` and `normal_theory_upper_limit` (Welch's *t* interval on the same mean difference, reported for contrast); `smd` with `smd_lower_limit` and `smd_upper_limit` from `ci_smd`; `cles` with `cles_lower_limit` and `cles_upper_limit` from `cles`; and `cliff_delta` with `cliff_delta_lower_limit` and `cliff_delta_upper_limit` from `cliff_delta`.

The design: `n_1`, `n_2`, `N`, `n_evaluated` (how many reassignments were actually used), and `exact` (1 if every reassignment was enumerated, 0 if they were sampled).

Non-numeric information travels on attributes rather than in the `value` column: `statistic_name`, `method` ("exact enumeration" or "Monte Carlo"), `alternative`, `group_labels`, `response_name`, `group_name`, `seed`, `observed_statistic`, and `reference_distribution`, the vector of statistics over the reassignments that `plot_randomization_test` draws.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Edgington, E. S., & Onghena, P. (2007). *Randomization tests* (4th ed.). Chapman & Hall/CRC.

Ernst, M. D. (2004). Permutation methods: A basis for exact inference. *Statistical Science, 19*(4), 676–685. doi:10.1214/088342304000000396

Fisher, R. A. (1935). *The design of experiments*. Oliver & Boyd.

Janssen, A. (1997). Studentized permutation tests for non-i.i.d. hypotheses and the generalized Behrens-Fisher problem. *Statistics & Probability Letters, 36*(1), 9–21. doi:10.1016/S01677152(97)00043-6

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 1 on the logic of randomization and the randomization test.)

Neuhaus, G. (1993). Conditional rank tests for the two-sample problem under random censorship. *The Annals of Statistics, 21*(4), 1760–1779. doi:10.1214/aos/1176349396

Phipson, B., & Smyth, G. K. (2010). Permutation *p*-values should never be zero: Calculating exact *p*-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology, 9*(1), Article 39. doi:10.2202/15446115.1585

Pitman, E. J. G. (1937). Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society, 4*(1), 119–130.

See Also

plot_randomization_test for the figure that shows the reference distribution, the observed statistic, and the rejection region; randomization_test_paired for the sign-flip sibling used with paired observations; t.test and wilcox.test for the parametric and rank-based alternatives; smd, ci_smd, cles, and cliff_delta for the effect sizes reported here.

Other hypothesis tests: adjusted_means(), ancova(), anova_within(), ci_dunnett(), ci_scheffe(), ci_tukey_kramer(), compare_cov_structures(), contrast_test(), correlations_test(), equivalence_r(), equivalence_smd(), factorial_anova(), manova_split_plot(), mauchly_test(), mixed_anova(), obrien_test(), pairwise_within(), randomization_test_paired(), regions_of_significance(), simple_effects_AB(), summary_t_test(), welch_t()

Examples

```
# 1. Ten observations, so every one of the choose(10, 5) = 252
#    reassignments is enumerated and the p-value is exact.
treatment <- c(80, 84, 79, 88, 83)
control   <- c(72, 75, 68, 81, 74)
randomization_test(group_1 = treatment, group_2 = control)

# 2. The studentized statistic, preferable when the groups may differ
#    in variance.
randomization_test(group_1 = treatment, group_2 = control,
                   statistic = "t")

# 3. Formula interface: weekly drinking in the two comparable arms of
#    the drinks_trial data, a right-skewed outcome, which is exactly
#    where a distribution-free test earns its keep. With 37 and 32
#    participants there are far too many reassignments to enumerate, so
#    10,000 are drawn and the p-value carries a Monte Carlo standard
#    error.
cra <- droplevels(subset(drinks_trial, treatment != "CRA + Disulfiram"))
set.seed(113)
randomization_test(drinks_per_week ~ treatment, data = cra, seed = 113)

# 4. A one-sided test, and the one-sided interval that goes with it.
randomization_test(group_1 = treatment, group_2 = control,
                   alternative = "greater")

# 5. On ranks, the test is the exact Wilcoxon rank sum test.
y <- c(treatment, control)
g <- rep(c("treatment", "control"), each = 5)
res <- randomization_test(rank(y), g)
res$value[res$term == "p_value"]
wilcox.test(treatment, control, exact = TRUE)$p.value
```

`randomization_test_paired`

Paired-Samples Randomization (Sign-Flip) Test

Description

Computes an exact (or Monte Carlo) sign-flip randomization test for paired observations (x_i, y_i), treating the within-pair sign of $d_i = y_i - x_i$ as the randomization mechanism (Fisher, 1971; Edgington & Onghena, 2007). Under the null hypothesis of exchangeability (H_0: the labeling of x and y within each pair is arbitrary), each of the 2^n sign patterns is equally likely.

Usage

```
randomization_test_paired(
  x,
  y,
  statistic = c("mean", "t"),
  alternative = c("two_sided", "less", "greater"),
  exact = NULL,
  n_resamples = 10000L,
  seed = NULL
)
```

Arguments

x, y	Paired numeric vectors of equal length. NAs are removed pairwise.
statistic	One of "mean" (default) or "t". "mean" uses the mean difference $\bar{d}$ as the test statistic. "t" uses the paired *t*-statistic $\bar{d}\sqrt{n}/s_d$, which is more robust to pair-to-pair variability in $\lvert d_i \rvert$.
alternative	One of "two_sided" (default; the base-R spelling "two.sided" is accepted as an alias), "less", or "greater".
exact	Logical. If NULL (default), uses exact enumeration when $n \leq 20$ and Monte Carlo otherwise. If TRUE, forces exact enumeration (caps at $n \leq 25$; above that the 2^n space is too large). If FALSE, forces Monte Carlo.
n_resamples	Number of Monte Carlo resamples when exact enumeration is not used. Default 10000L.
seed	Optional integer seed for reproducibility of the Monte Carlo branch. Default NULL, which leaves the user's current RNG state intact; supply an integer for reproducibility.

Details

For small n (default $n \leq 20$) the test enumerates all 2^n sign patterns exactly; for larger n a Monte Carlo approximation is used (default n_resamples = 10000L).

Why randomization. The randomization test makes no distributional assumption on d_i; it only assumes that under the null, the sign of each d_i is arbitrary. This is exactly the inference that pre-experimental random assignment licenses, and it is robust to heavy-tailed differences, mixtures, and outliers.

Exact enumeration. For $n \leq 25$, all 2^n sign patterns are enumerated. The observed test statistic is compared with the full reference distribution. The exact two-sided *p*-value is the proportion of patterns yielding a test statistic at least as extreme (in absolute value) as the observed.

Monte Carlo branch. For larger n, `n_resamples` random sign patterns are drawn uniformly from $\{-1, +1\}^n$; the Monte Carlo p-value uses the standard $(1 + \text{count}) / (1 + B)$ plug-in to avoid $p = 0$.

Value

A `data.frame` with rows for the observed test statistic, the p-value, the number of pairs, the number of randomizations evaluated, and a flag indicating whether the test was exact or Monte Carlo.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Edgington, E. S., & Onghena, P. (2007). *Randomization tests* (4th ed.). Chapman & Hall/CRC.

Fisher, R. A. (1971). *The design of experiments* (9th ed., reprint). Hafner.

Pitman, E. J. G. (1937). Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society, 4*(1), 119–130.

See Also

`t.test` (parametric paired test), `probability_of_superiority_paired`

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `correlations_test()`, `equivalence_r()`, `equivalence_smd()`, `factorial_anova()`, `manova_split_plot()`, `mauchly_test()`, `mixed_anova()`, `obrien_test()`, `pairwise_within()`, `randomization_test()`, `regions_of_significance()`, `simple_effects_AB()`, `summary_t_test()`, `welch_t()`

Examples

```
# 1. Small-n exact: Bayley scores on twin pairs.
control <- c(95, 102,  98, 107, 105)
treat   <- c(102, 108, 100, 112, 109)
randomization_test_paired(control, treat)

# 2. Larger n: Monte Carlo branch.
set.seed(113)
x <- rnorm(50, 100, 15)
y <- x + rnorm(50,   5, 12)
randomization_test_paired(x, y, n_resamples = 10000L)
```

`regions_of_significance` *Regions of Significance for a Covariate by Group Interaction*

Description

Finds the values of a covariate at which two groups differ significantly when the within-group regression slopes are *not* equal, that is, when there is a covariate-by-group interaction (heterogeneity of regression). With more than two groups the calculation is carried out for every pair of groups.

Usage

```
regions_of_significance(
  object,
  data = NULL,
  conf_level = 0.95,
  method = c("simultaneous", "pointwise")
)
```

Arguments

`object`	Either a fitted `lm` or `aov` of the form `y ~ x * group`, or a model formula of that form. When a formula is supplied, `data` must be supplied too and the model is fit with `lm`.
`data`	A `data.frame` holding the variables in the formula. Used, and required, only when `object` is a formula; ignored with a warning when `object` is already a fitted model.
`conf_level`	Confidence level for the boundaries. Default `0.95`.
`method`	Character string naming the critical value. `"simultaneous"` (the default) uses Potthoff's (1964) simultaneous critical value $\sqrt{2F_{1-\alpha;2,\nu}}$, which holds the error rate over the whole covariate continuum at once. `"pointwise"` uses the classic critical value $t_{1-\alpha/2;\nu}$, which holds it at one covariate value chosen in advance.

Details

Why the procedure exists. An analysis of covariance that assumes a common within-group slope reports one adjusted mean difference, and that single number is a complete summary of the group comparison only if the slopes really are common. When the covariate interacts with the group factor the slopes are not common, the two fitted lines converge or cross, and there is no such thing as "the" treatment effect: the difference between the groups depends on where along the covariate you look. Reporting the adjusted mean difference anyway reports the difference at one covariate value, the covariate grand mean, and says nothing about the rest of the range. The question worth answering is instead *where* on the covariate the groups differ, and that is what this function answers.

The calculation. For two groups, write the estimated difference at covariate value x as the line

$$\hat{D}(x) = \hat{d}_0 + \hat{d}_1 x,$$

where $\hat{d}_0$ is the difference in intercepts and $\hat{d}_1$ the difference in slopes. Because $\hat{D}(x)$ is a linear combination of the regression coefficients, its sampling variance follows from their covariance matrix,

$$\mathrm{Var}[\hat{D}(x)] = \mathrm{Var}(\hat{d}_0) + 2x\,\mathrm{Cov}(\hat{d}_0, \hat{d}_1) + x^2\mathrm{Var}(\hat{d}_1).$$

The groups differ significantly at x exactly when $\hat{D}(x)^2 > t^2_{crit}\,\mathrm{Var}[\hat{D}(x)]$. Setting the two sides equal gives a quadratic in x,

$$(\hat{d}_1^2 - t^2_{crit}\mathrm{Var}(\hat{d}_1))x^2 + 2(\hat{d}_0\hat{d}_1 - t^2_{crit}\mathrm{Cov}(\hat{d}_0, \hat{d}_1))x + (\hat{d}_0^2 - t^2_{crit}\mathrm{Var}(\hat{d}_0)) = 0,$$

whose real roots are the boundaries of the region of significance.

Every geometry is possible, and the leading coefficient decides which. The coefficient on x^2 is positive exactly when the slope difference itself clears the critical value. When it is positive the parabola opens upward and the groups differ *outside* the two boundaries, the familiar picture of two lines that cross somewhere in the middle of the covariate and separate at both ends. When it is negative the parabola opens downward and the groups differ *between* the boundaries, a middle band of covariate values where the two lines are far enough apart relative to the precision available there. When there are no real roots the sign never changes, so the groups differ either everywhere or nowhere. All of these are reported through `region_code` rather than being treated as failures.

Of those, "everywhere" is a case the code enumerates but the mathematics rules out whenever the slopes genuinely differ. The estimated difference $\hat{D}(x)$ is then a nonconstant line in x, so it crosses zero at some covariate value, and at that value the squared difference is zero while the critical bound is positive. The quadratic therefore always has two real roots, and the significant set is the pair of tails outside them or the band between them, never the whole line. Note this is a statement about the covariate axis extended without limit, not about the observed data: within the range actually observed the groups may well differ everywhere, which is why the next paragraph matters.

Boundaries outside the observed data. A boundary is a root of an equation, and the equation is happy to place it far outside the covariate values that were actually observed. Such a boundary is an extrapolation of two fitted lines into a region where there are no data to support them, and it should not be read as a covariate value at which anything can be claimed. The boundary is still reported, since suppressing it would hide the shape of the result, but it is flagged in `lower_bound_in_range` and `upper_bound_in_range` and noted in the `region` string.

Simultaneous versus pointwise. The classic critical value is $t_{1-\alpha/2}$, which controls the Type I error rate at a *single* covariate value fixed in advance. That is not what anyone actually does: the whole point of the procedure is to scan the covariate continuum and read off where the groups differ, which is a search over infinitely many tests. Potthoff (1964) gave the simultaneous critical value $\sqrt{2F_{1-\alpha;2,\nu}}$, which holds the error rate over the entire covariate range at once and is therefore the default here, and the value used in Maxwell, Delaney, and Kelley (2027, Chapter 9). The simultaneous critical value is always the larger of the two, so its region of significance is always the more conservative one. With more than two groups the simultaneous guarantee applies to each pair over the covariate range, not to the family of pairs. For a Bonferroni protection across the pairs, divide the Type I error rate by the number of pairs before choosing `conf_level`: with three groups (three pairs) and a familywise rate of .05, pass `conf_level = 1 - .05 / 3`.

What the model may contain. The model must contain exactly one interaction between a numeric covariate and a grouping factor, and the grouping factor may not appear in any other term. Additional predictors that do not interact with the group are allowed and drop out of the group difference, so `y ~ x * group + block` is fine while `y ~ x * group * block` is not. A grouping variable stored as a number (0/1, say) is a numeric predictor to R, not a factor, so convert it with `factor` first.

Value

A `data.frame` (class `dmar_tbl`) with one row per reported quantity per pair of groups and columns `pair`, `term`, and a numeric `value`. The terms, for each pair, are

`lower_bound`, `upper_bound` The boundaries of the region, sorted, on the scale of the covariate. `NA` when a boundary does not exist; when there is a single boundary it is reported as `lower_bound` and `upper_bound` is `NA`. Read them with `region_code`: the boundaries are the covariate values at which the group difference sits exactly on the critical value, and it is `region_code` that says on which side of them the groups differ.

`region_code` How to read the boundaries. `1`: two boundaries, the groups differ *outside* them. `2`: two boundaries, the groups differ *between* them. `3`: no boundary, the groups differ at every covariate value. `4`: no boundary, the groups differ at no covariate value. `5` and `6`: one boundary, the groups differ above it (`5`) or below it (`6`).

`n_boundaries` How many real boundaries exist: 0, 1, or 2.

`lower_bound_in_range`, `upper_bound_in_range` `1` when the boundary falls inside the covariate values actually observed in the two groups, `0` when it falls outside them, `NA` when the boundary does not exist. A boundary outside the observed range is an extrapolation of the fitted lines and should not be interpreted as a covariate value at which anything was or could be seen.

`difference_intercept`, `difference_slope` The intercept d_0 and slope d_1 of the group difference as a function of the covariate, so that the estimated difference at covariate value x is $d_0 + d_1 x$.

`critical_value` The critical value used, t_{crit}.

`df_error` Error degrees of freedom of the fitted model.

`conf_level` The confidence level.

Everything that is not a number is carried on attributes rather than forced into `value`: `method`, `outcome`, `covariate`, `group`, the overall observed `covariate_range`, a `pairs` `data.frame` with the group labels, the per-pair observed covariate range, and a plain-language `region` string for each pair, and a `geometry` `data.frame` holding d_0, d_1, and the three elements of their covariance matrix, which is what `plot_regions_of_significance` draws.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Johnson, P. O., & Neyman, J. (1936). Tests of certain linear hypotheses and their application to some educational problems. *Statistical Research Memoirs, 1*, 57–93.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9 and its extension on heterogeneity of regression.)

Potthoff, R. F. (1964). On the Johnson-Neyman technique and some extensions thereof. *Psychometrika, 29*(3), 241–256. doi:10.1007/BF02289721

Rogosa, D. (1980). Comparing nonparallel regression lines. *Psychological Bulletin, 88*(2), 307–321. doi:10.1037/00332909.88.2.307

See Also

plot_regions_of_significance to see the group difference and its confidence band across the covariate; ancova for the common-slope analysis and its homogeneity-of-regression test; pygmalion for the data used below.

Other hypothesis tests: adjusted_means(), ancova(), anova_within(), ci_dunnett(), ci_scheffe(), ci_tukey_kramer(), compare_cov_structures(), contrast_test(), correlations_test(), equivalence_r(), equivalence_smd(), factorial_anova(), manova_split_plot(), mauchly_test(), mixed_anova(), obrien_test(), pairwise_within(), randomization_test(), randomization_test_paired(), simple_effects_AB(), summary_t_test(), welch_t()

Examples

```
# ---- The Pygmalion teacher-expectancy data ----
# Post-test IQ, averaged over the two follow-up assessments, on
# pretest IQ, separately by condition. The slopes differ, so the
# expectancy effect depends on where the child started.
data(pygmalion)
pygmalion$iq_post <- (pygmalion$iq_4 + pygmalion$iq_8) / 2
fit <- lm(iq_post ~ iq_pre * treatment, data = pygmalion)

regions_of_significance(fit)

# The plain-language reading of each pair is on an attribute.
attr(regions_of_significance(fit), "pairs")$region

# The pointwise (classic) critical value gives a wider region,
# because it does not pay for scanning the whole covariate.
regions_of_significance(fit, method = "pointwise")

# ---- Formula interface, and more than two groups ----
set.seed(113)
n <- 150
g <- factor(rep(c("control", "low", "high"), each = n / 3))
x <- rnorm(n, 50, 10)
y <- 2 + 0.5 * x + (g == "high") * (0.4 * x - 15) + rnorm(n, 0, 5)
d <- data.frame(y, x, g)
regions_of_significance(y ~ x * g, data = d)
```

---

reliability *Reliability Coefficient With a Confidence Interval (General Dispatch)*

Description

General-purpose entry point for the reliability family. Dispatches to reliability_alpha, reliability_kr20, reliability_omega, or reliability_omega_categorical according to the requested type. When type is not specified, the function picks a reasonable default from the supplied input following the recommendations of Kelley and Pornprasertmanit (2016).

Usage

```
reliability(
  data = NULL,
  S = NULL,
  N = NULL,
  type = NULL,
  estimator = c("analytic", "model_implied"),
  denominator = c("observed", "model_implied"),
  missing = c("listwise", "fiml"),
  aux = NULL,
  ci_method = NULL,
  conf_level = 0.95,
  B = 10000,
  seed = NULL
)
```

Arguments

`data`	A numeric matrix or data frame of item scores, or `NULL`.
`S`	A symmetric covariance matrix among the items, or `NULL`. If supplied, `N` must also be supplied; methods that require raw data are then unavailable.
`N`	Total sample size; required when `S` is supplied.
`type`	Character; one of `"alpha"`, `"kr20"`, `"omega"`, `"omega_categorical"` (`"omega_c"` is accepted as a shorthand), or `NULL` for auto-detection. See *Details*.
`estimator`	For `type = "alpha"` only: how coefficient α is estimated, `"analytic"` (default; the closed-form equation applied to the observed covariance matrix) or `"model_implied"` (the reliability implied by the τ-equivalent single-factor model fit by maximum likelihood); forwarded to `reliability_alpha`, whose help page discusses the choice and which interval methods each estimator supports. Supplying it with any other `type` is an error, as `denominator` is for any type but `"omega"`.
`denominator`	For `type = "omega"` only: how the total variance in the denominator of ω is estimated, `"observed"` (default; robust omega) or `"model_implied"`; forwarded to `reliability_omega`, whose help page discusses the choice.
`missing`	For `type = "alpha"` and `type = "omega"` only: how incomplete rows of `data` are handled, `"listwise"` (the default) or `"fiml"` (full information maximum likelihood); forwarded to the family function, whose help page discusses the choice. Supplying it with any other `type` is an error.
`aux`	For `type = "alpha"` and `type = "omega"` only: optional character vector naming auxiliary variable columns of `data`, entered as saturated correlates under full information maximum likelihood; forwarded to the family function. Supplying `aux` implies `missing = "fiml"`.
`ci_method`	Method for constructing the confidence interval, or `NULL` to use the chosen family function's default. See the help page for the chosen `reliability_*` function for the full list of accepted values.
`conf_level`	Confidence level. Defaults to `0.95`.

B	Number of bootstrap replications when a bootstrap method is selected. Defaults to 10000.
seed	Random number seed used for bootstrap reproducibility. Defaults to NULL, which leaves the user's current RNG state intact; supply an integer for reproducibility.

Details

Auto-detection rules (used only when type = NULL):

- If raw items are integer-valued and every column has at most 10 distinct values, type = "omega_categorical" (categorical omega; appropriate when items are ordered categorical and the relationship between the underlying factor and the observed items is non-linear).
- Otherwise, type = "omega" (McDonald's coefficient ω from a single-factor CFA). This includes the case where only a covariance matrix S and sample size N are supplied; **lavaan** fits the CFA on the covariance matrix.

Auto-detection emits a single message() indicating which type was chosen, so it never surprises the user silently.

The selected family function determines which ci_method values are accepted; see the help page for the chosen function for the full list. When ci_method is left at its default (NULL), the family function's own default is used:

- reliability_alpha: "bonett".
- reliability_alpha(estimator = "model_implied"): "mlr" with raw data; "ml" with covariance input.
- reliability_kr20: "feldt".
- reliability_omega: for denominator = "observed" (robust omega, the default), the point estimate with no interval, since its interval is bootstrap based and no bootstrap runs unless requested; for denominator = "model_implied", "mlr".
- reliability_omega_categorical: the point estimate with no interval, for the same reason; request "bca".

A bootstrap is never run by default anywhere in the family. When a bootstrap method is requested, B = 10000 replications is the default.

Several reliability coefficients exist because their assumptions differ. Coefficient α (and its dichotomous specialization KR-20) equals the population reliability under essential τ-equivalence (equal loadings); McDonald's ω relaxes that assumption to a congeneric single-factor model; and reliability_omega(denominator = "observed") further relaxes the requirement that the single-factor model fit perfectly by using the observed composite variance in the denominator (see the reliability_omega help page for the properties of that choice). For well-behaved homogeneous measurement instruments these coefficients typically yield very similar values. For ordered categorical items the relationship between the latent factor and the observed responses is non-linear, and reliability_omega_categorical handles that case explicitly via a probit-link single-factor model.

Value

The data.frame returned by the dispatched reliability_* function (rows: estimate, se, lower_limit, upper_limit, conf_level, N, N_complete, J). The se row is on the coefficient scale; the transformation-based intervals ("fisher", "bonett", "hakstian_whalen") add an se_transformed row carrying the transformation-scale standard error, with the scale named in the se_transform_scale attribute. The coefficient attribute identifies which coefficient was computed, and a bootstrap call also carries the bootstrap attribute with its replication counts (see reliability_alpha).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Cheng, Y. (2012). Estimation of and confidence interval formation for reliability coefficients of homogeneous measurement instruments. *Methodology, 8*, 39–50. doi:10.1027/1614-2241/a000036

Kelley, K., & Pornprasertmanit, S. (2016). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for composite measures. *Psychological Methods, 21*, 69–92. doi:10.1037/a0040086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Terry, L. J., & Kelley, K. (2012). Sample size planning for composite reliability coefficients: Accuracy in parameter estimation via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 371–401. doi:10.1111/j.20448317.2011.02030.x

See Also

reliability_alpha, reliability_kr20, reliability_omega, reliability_omega_categorical, ss_aipe_reliability, cfa_1.

Other reliability: cohen_kappa(), diagnosis_agreement, fleiss_kappa(), icc(), reliability_H(), reliability_alpha(), reliability_kr20(), reliability_omega(), reliability_omega_categorical()

Examples

```
set.seed(113)
J <- 6
loadings <- seq(0.4, 0.8, length.out = J)
eta <- rnorm(200)
errors <- matrix(rnorm(200 * J), 200, J) %*% diag(sqrt(1 - loadings^2))
items <- sweep(matrix(rep(eta, J), 200, J), 2, loadings, `*`) + errors
colnames(items) <- paste0("y", seq_len(J))

# Auto-detection picks coefficient omega for continuous data.
reliability(data = items)

# Explicit type.
reliability(data = items, type = "alpha")
```

```
# A covariance matrix and its sample size stand in for raw data, and
# auto-detection again picks coefficient omega. The point estimate is
# the same as from the raw data; the message explains why no interval
# accompanies it, since the bootstrap behind robust omega resamples
# rows that a covariance matrix does not carry.
reliability(S = cov(items), N = 200)
```

`reliability_alpha` *Coefficient Alpha With a Confidence Interval*

Description

Estimates coefficient α for a homogeneous composite score and returns a confidence interval for the population coefficient using one of several documented methods.

Usage

```
reliability_alpha(
  data = NULL,
  S = NULL,
  N = NULL,
  estimator = c("analytic", "model_implied"),
  missing = c("listwise", "fiml"),
  aux = NULL,
  ci_method = NULL,
  conf_level = 0.95,
  B = 10000,
  seed = NULL
)
```

Arguments

`data`	A numeric matrix or data frame of item scores (rows are respondents, columns are items). Either `data` or `S` must be supplied. How incomplete rows are handled is governed by `missing`: listwise deletion by default, or full information maximum likelihood with `missing = "fiml"`. When `aux` is supplied, `data` contains both the item columns and the auxiliary columns, and the columns not named in `aux` are the items.
`S`	A symmetric covariance matrix among the items. If `S` is supplied, `N` must also be supplied; raw-data-only confidence interval methods (`"adf"`, `"bootstrap_*"`, `"percentile"`, `"bca"`) are then unavailable, as are `missing = "fiml"` and `aux`, since a covariance matrix has no incomplete cases.
`N`	Total sample size; required when `S` (rather than `data`) is supplied.
`estimator`	How α is estimated: `"analytic"` (the default) applies the closed-form equation to the observed covariance matrix, and `"model_implied"` takes the reliability implied by the τ-equivalent single-factor model fit by maximum likelihood. See *Details*.

`missing` How incomplete rows of `data` are handled: `"listwise"` (the default; complete-case analysis, the historical behavior) or `"fiml"` (full information maximum likelihood, using every case with at least one observed item). See *Details*.

`aux` Optional character vector naming auxiliary variable columns of `data`, entered as saturated correlates under full information maximum likelihood. Supplying `aux` implies `missing = "fiml"`. See *Details*.

`ci_method` Method for constructing the confidence interval. See *Details* for which methods each estimator supports. The default `NULL` resolves to `"bonett"` for the analytic estimator and to `"mlr"` for the model implied estimator (or `"ml"` when only a covariance matrix is supplied, since `"mlr"` needs raw data). With `missing = "fiml"` the analytic default is `"ml"`, whose standard error comes from the FIML information matrix.

`conf_level` Confidence level for the interval (1 - Type I error rate). Defaults to `0.95`.

`B` Number of bootstrap replications when a bootstrap method is selected. Defaults to `10000`.

`seed` Random number seed used for bootstrap reproducibility. Defaults to `NULL`, which leaves the user's current RNG state intact; supply an integer for reproducibility. When set, the seed applies inside the function only, and the user's generator state is restored when the function returns.

Details

Coefficient α was first derived by Guttman (1945) as his λ_3 and was subsequently popularized by Cronbach (1951), under whose name the coefficient is often informally cited. The attribution to Cronbach is historically incomplete; the modern literature increasingly refers to the coefficient simply as "coefficient alpha" (see, e.g., Sijtsma, 2009; Revelle & Zinbarg, 2009). DMAR follows that convention.

For a J-item composite $Y = \sum_j X_j$ the population coefficient is

$$\alpha = \frac{J}{J-1}\left(1 - \frac{\sum_j \sigma_j^2}{\sigma_Y^2}\right),$$

where σ_j^2 is the variance of item j and σ_Y^2 is the variance of the composite. The sample estimate substitutes sample variances. Under classical test theory, α equals the population reliability of the composite when the items are essentially τ-equivalent (i.e., equal factor loadings); when loadings differ, α is a lower bound on the population reliability. For well-behaved homogeneous measurement instruments coefficient α and McDonald's (1999) coefficient ω (`reliability_omega`) typically yield very similar values; ω extends to congeneric items (heterogeneous loadings) without the lower-bound caveat. For ordered categorical items see `reliability_omega_categorical`.

Two estimators of the same coefficient. The argument `estimator` selects how α is estimated from the data. Both target the same population quantity, and they agree in the population whenever the τ-equivalent model holds; they differ in a finite sample because they take different routes to it.

`"analytic"` The default. The closed-form equation above, applied to the observed covariance matrix. This is the classical coefficient, the number a hand calculation produces, and it makes no assumption beyond those of classical test theory.

`"model_implied"` The reliability implied by the τ-equivalent (equal loadings) single-factor model fit by maximum likelihood. With a shared loading λ and error variances ψ_j^2 the model implied reliability of the J-item composite is

$$\alpha = \frac{(J\lambda)^2}{(J\lambda)^2 + \sum_j \psi_j^2},$$

with maximum likelihood estimates substituted. Estimating through the model brings inference the formula cannot provide: a delta method standard error, a robust (Satorra-Bentler) variant under nonnormality, the profile likelihood interval, and a fit assessment of the τ-equivalence claim itself. When the equal-loadings claim is doubtful, the congeneric `reliability_omega` is the appropriate coefficient rather than either α.

Users of **MBESS** will recognize these as `ci.reliability(type = "alpha")` and `ci.reliability(type = "alpha-cfa")` respectively.

Missing data and auxiliary variables. By default incomplete rows are listwise-deleted, which is unbiased only when the data are missing completely at random and is inefficient always; the `mlmr` vignette develops the argument at length. Setting `missing = "fiml"` keeps every case with at least one observed item and estimates by full information maximum likelihood, which is consistent and efficient under the weaker missing at random (MAR) assumption. The `aux` argument names auxiliary variables: columns of `data` that are not part of the composite but are correlated with the items or with the reasons values are missing. They are entered as *saturated correlates* (Graham, 2003): correlated freely with each other and with every item's residual, never loading on the factor and never entering the composite, so the measurement model is undisturbed while FIML uses their information. Beyond recovering information, a good auxiliary makes the MAR assumption itself more plausible, since missingness that depends on the auxiliary becomes MAR once the auxiliary is conditioned on (Collins, Schafer, & Kam, 2001). Supplying `aux` implies `missing = "fiml"`; combining it with an explicit `missing = "listwise"` is an error. Listwise deletion remains the default so no existing result changes and the missing-data treatment is always a visible, deliberate choice. How each estimator uses FIML: the model implied estimator simply fits its model with `missing = "ml"`; the analytic estimator applies the classical formula, unchanged, to the FIML estimate of the item covariance matrix (from a saturated model over the items and any auxiliaries), so the estimand stays the classical coefficient and only the covariance matrix it is computed from improves. Under `missing = "fiml"` the available intervals are `"ml"` and `"ml_logistic"` (both estimators; the standard error comes from the FIML information matrix), `"mlr"` and `"mlr_logistic"` (model implied estimator; the Yuan-Bentler robust standard error), and the bootstrap methods, which resample rows (including the partially observed ones) and refit by FIML on each replication. The complete-data closed forms (`"feldt"`, `"fisher"`, `"bonett"`, `"hakstian_whalen"`), `"adf"`, and `"likelihood"` are errors with `missing = "fiml"` rather than silently reverting to listwise deletion. Multiple imputation is a different feature with a different interface and is out of scope here.

Available confidence interval methods (set via `ci_method`). Some belong to one estimator only, because a closed-form interval for the sample coefficient and a model-based interval are not interchangeable; requesting a method the chosen estimator cannot supply is an error that names the estimator to use instead:

`"feldt"` *Analytic estimator only.* The *F*-distribution interval of Feldt (1965), exact under multivariate normality and parallel items.

`"fisher"` *Analytic estimator only.* Fisher's z' transformation (Fisher, 1950). Tends to overcover (Padilla, Divers, & Newton, 2012) and is generally not recommended.

`"bonett"` *Analytic estimator only.* Bonett's (2002) log transformation. The default for that estimator; well-behaved under normality.

`"hakstian_whalen"` *Analytic estimator only.* Cube-root transformation of Hakstian and Whalen (1976).

`"mlr"`, `"mlr_logistic"` *Model implied estimator only.* Wald interval using the robust (Satorra & Bentler, 1994) standard error from the fitted model. The default for that estimator; recommended among the closed forms when item distributions deviate from normality (Kelley & Pornprasertmanit, 2016). Requires raw data.

`"likelihood"` *Model implied estimator only.* Profile likelihood interval: the set of population values not rejected by the likelihood ratio test under the τ-equivalent model, located by refitting under the nonlinear constraint that the model implied reliability equals each candidate value. Respects [0, 1], is not forced to be symmetric, and works from raw data or covariance input. Requires **lavaan**. It is unavailable with the analytic estimator because the interval and the point estimate would then refer to different quantities: the interval profiles the model implied coefficient while the estimate is the sample coefficient, so under a misspecified model the interval can exclude the estimate it accompanies.

`"ml"`, `"ml_logistic"` Available to both estimators, by the route each affords. With `"analytic"` it is the closed-form ML standard error of van Zyl, Neudecker, and Nel (2000), computed directly from the covariance matrix; with `"model_implied"` it is the delta method standard error from the fitted model. The `_logistic` variant applies Browne's (1982) logit transformation.

`"adf"`, `"adf_logistic"` Available to both estimators. With `"analytic"` it is the asymptotic distribution-free standard error of Maydeu-Olivares, Coffman, and Hartmann (2007); with `"model_implied"` the model is fit by weighted least squares (Browne, 1984) and the delta method applied. Requires raw data and a relatively large sample size.

`"bootstrap_se"`, `"bootstrap_se_logistic"`, `"percentile"`, `"bca"` Available to both estimators. Nonparametric bootstrap intervals (Efron & Tibshirani, 1993): the rows of `data` are resampled with replacement `B` times and the chosen estimator is recomputed on each replication. `"percentile"` takes the interval limits from the empirical quantiles of the bootstrap estimates (for a 95 percent interval, the 2.5th and 97.5th percentiles); it respects [0, 1] and is not forced to be symmetric about the estimate, but its coverage degrades when the estimator is biased or its variance changes with the parameter. `"bca"` (bias-corrected and accelerated) adjusts the two quantile positions for exactly those features, estimating the median bias from the bootstrap distribution and the acceleration from the jackknife, and is second-order accurate where the percentile interval is first-order accurate (DiCiccio & Efron, 1996). The acceleration is estimated as **boot** estimates it, by regression on the resampling frequencies when at least as many replications as cases returned a value and by the jackknife otherwise; a leave-one-out refit that fails is counted and set aside rather than stopping the call (as one did through version 1.0.0), and when the acceleration or the bias correction cannot be estimated at all the percentile interval is returned in its place, with a warning and the substitution recorded in the `ci_method` attribute; see `reliability_omega_categorical` for the full account. `"bootstrap_se"` uses the standard deviation of the bootstrap estimates as a standard error in an ordinary normal-theory interval; the `_logistic` variant builds that interval on the logit scale so the endpoints respect [0, 1]. Replications on which the estimator cannot be computed (for example, a model refit that does not converge) are set aside, and the interval is computed from the replications that return a value, whose quantiles the percentile and BCa limits are; the counts, and why the rest were set aside, travel in the `bootstrap` attribute (see *Value*), print under the table, and are stated in a warning whenever a replication was set aside. Requires raw data and the

boot package. The default `B = 10000` is an accuracy choice: the BCa adjustment pushes the working quantiles farther into the tails of the bootstrap distribution than the percentile interval uses, and stabilizing them takes more replications than the customary 2000; reduce `B` for exploration, not for a reported analysis. Bootstrap results vary from run to run; supply `seed` for reproducibility.

`"none"` Return only the point estimate.

Comparison with other packages. The **psych** package provides `alpha`, which reports coefficient α along with item-level diagnostics (item-total correlations, alpha-if-item-deleted, and several alternative coefficients). The emphasis in **psych** is broad exploratory psychometric reporting. `reliability_alpha` in **DMAR** differs in emphasis: it returns a single point estimate alongside a principled confidence interval drawn from the methods compared in Kelley and Pornprasertmanit (2016).

Value

A `data.frame` with columns `term` and `value` and rows `"estimate"` (sample coefficient α), `"se"` (standard error on the coefficient scale, `NA` for methods that do not produce one; for the transformation-based intervals `"fisher"`, `"bonett"`, and `"hakstian_whalen"` it is the delta method back-transform of the transformation-scale standard error, evaluated at the estimate), `"se_transformed"` (only for those transformation-based intervals: the standard error on the transformation scale, the quantity the interval is built from, with the scale named in the attribute `se_transform_scale`: `"fisher_z"`, `"log(1-alpha)"`, or `"cube_root"`), `"lower_limit"` and `"upper_limit"` (clamped to [0, 1]), `"conf_level"`, `"N"` (the cases the analysis used: the complete cases under listwise deletion, every case with at least one observed item under `missing = "fiml"`), `"N_complete"` (the complete cases, so the cost of listwise deletion is visible at a glance; equal to `"N"` under listwise deletion and for covariance input), and `"J"` (number of items). The selected coefficient, CI method, and missing-data treatment travel as the attributes `coefficient`, `ci_method`, `missing`, and (when supplied) `aux`; bootstrap calls also record `B`. A bootstrap call also carries the attribute `bootstrap`, a list recording the replications the interval conditions on: `attempted` (`B`), `returned` (the replications that returned a finite value; the standard error is their standard deviation and the percentile and BCa limits are their quantiles), `failed` (the difference), `reasons` (a named count of why replications were set aside: `error` when the refit raised an error, `non_finite` when it returned no finite value, as a fit that did not converge does), `error_messages` (the distinct error messages with their counts, empty when none), `acceleration` and `acceleration_method` (the BCa acceleration used and whether it came from the regression or the jackknife; `NA` unless a BCa interval was formed), and `jackknife` (the same `attempted`, `returned`, `failed`, `reasons`, and `error_messages` for the leave-one-out refits behind the acceleration, all zero when no jackknife ran). The printed table shows these counts in one line, and a warning states them whenever a replication was set aside.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bonett, D. G. (2002). Sample size requirements for testing and estimating coefficient alpha. *Journal of Educational and Behavioral Statistics, 27*(4), 335–340. doi:10.3102/10769986027004335

Browne, M. W. (1982). Covariance structures. In D. M. Hawkins (Ed.), *Topics in applied multivariate analysis* (pp. 72–141). Cambridge, UK: Cambridge University Press.

Browne, M. W. (1984). Asymptotically distribution-free methods for the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology, 37*, 62–83.

Collins, L. M., Schafer, J. L., & Kam, C.-M. (2001). A comparison of inclusive and restrictive strategies in modern missing data procedures. *Psychological Methods, 6*, 330–351. doi:10.1037/1082989X.6.4.330

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika, 16*(3), 297–334.

DiCiccio, T. J., & Efron, B. (1996). Bootstrap confidence intervals. *Statistical Science, 11*(3), 189–228.

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, NY: Chapman & Hall/CRC.

Feldt, L. S. (1965). The approximate sampling distribution of Kuder-Richardson reliability coefficient twenty. *Psychometrika, 30*, 357–370.

Fisher, R. A. (1950). *Statistical methods for research workers*. Edinburgh, UK: Oliver & Boyd.

Graham, J. W. (2003). Adding missing-data-relevant variables to FIML-based structural equation models. *Structural Equation Modeling, 10*(1), 80–100. doi:10.1207/S15328007SEM1001_4

Guttman, L. (1945). A basis for analyzing test-retest reliability. *Psychometrika, 10*(4), 255–282.

Hakstian, A. R., & Whalen, T. E. (1976). A *k*-sample significance test for independent alpha coefficients. *Psychometrika, 41*, 219–231.

Kelley, K. (2007a). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K. (2007b). Methods for the behavioral, educational, and social sciences: An R package. *Behavior Research Methods, 39*(4), 979–984. doi:10.3758/BF03192993

Kelley, K., & Cheng, Y. (2012). Estimation of and confidence interval formation for reliability coefficients of homogeneous measurement instruments. *Methodology, 8*, 39–50. doi:10.1027/1614-2241/a000036

Kelley, K., & Pornprasertmanit, S. (2016). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for composite measures. *Psychological Methods, 21*, 69–92. doi:10.1037/a0040086

Satorra, A., & Bentler, P. M. (1994). Corrections to test statistics and standard errors in covariance structure analysis. In A. von Eye & C. C. Clogg (Eds.), *Latent variables analysis: Applications for developmental research* (pp. 399–419). Thousand Oaks, CA: Sage.

Yuan, K.-H., & Bentler, P. M. (2000). Three likelihood-based methods for mean and covariance structure analysis with nonnormal missing data. *Sociological Methodology, 30*, 165–200.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Maydeu-Olivares, A., Coffman, D. L., & Hartmann, W. M. (2007). Asymptotically distribution-free (ADF) interval estimation of coefficient alpha. *Psychological Methods, 12*, 157–176. doi:10.1037/1082989X.12.2.157

McDonald, R. P. (1999). *Test theory: A unified treatment*. Mahwah, NJ: Lawrence Erlbaum Associates.

Padilla, M. A., Divers, J., & Newton, M. (2012). Coefficient alpha bootstrap confidence interval under nonnormality. *Applied Psychological Measurement, 36*, 331–348. doi:10.1177/0146621612445470

Revelle, W., & Zinbarg, R. E. (2009). Coefficients alpha, beta, omega, and the GLB: Comments on Sijtsma. *Psychometrika, 74*, 145–154. doi:10.1007/s113360089102z

Sijtsma, K. (2009). On the use, the misuse, and the very limited usefulness of Cronbach's alpha. *Psychometrika, 74*, 107–120. doi:10.1007/s1133600891010

Terry, L. J., & Kelley, K. (2012). Sample size planning for composite reliability coefficients: Accuracy in parameter estimation via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 371–401. doi:10.1111/j.20448317.2011.02030.x

van Zyl, J. M., Neudecker, H., & Nel, D. G. (2000). On the distribution of the maximum likelihood estimator of Cronbach's alpha. *Psychometrika, 65*(3), 271–280. doi:10.1007/BF02296146

See Also

`reliability` (general wrapper that dispatches by coefficient), `reliability_omega`, `reliability_omega_categorical`, `reliability_kr20`, `cfa_1` (single-factor CFA used internally), `ss_aipe_reliability`, `alpha`.

Other reliability: `cohen_kappa()`, `diagnosis_agreement`, `fleiss_kappa()`, `icc()`, `reliability()`, `reliability_H()`, `reliability_kr20()`, `reliability_omega()`, `reliability_omega_categorical()`

Examples

```
set.seed(113)
# Simulate six tau-equivalent items with population reliability ~ .8.
J <- 6
loadings <- rep(0.6, J)
eta <- rnorm(200)
errors <- matrix(rnorm(200 * J, sd = sqrt(1 - 0.6^2)), 200, J)
items <- outer(eta, loadings) + errors
colnames(items) <- paste0("y", seq_len(J))

# Default (Bonett's transformation) CI from raw data.
reliability_alpha(data = items)

# Same point estimate from a covariance matrix; CI requires N.
S <- cov(items)
reliability_alpha(S = S, N = 200, ci_method = "feldt")

# The bootstrap intervals resample the rows and recompute the
# coefficient on each replication. The percentile interval reads its
# limits off the empirical quantiles of the bootstrap estimates.
# B = 500 keeps the example quick; a reported interval deserves the
# default B = 10000, and seed makes the interval reproducible.
reliability_alpha(data = items, ci_method = "percentile", B = 500,
                  seed = 113)

# The bias-corrected and accelerated interval adjusts those two
# quantile positions for median bias and for acceleration, and is the
# better choice for a reported interval. The default B = 10000 is an
# accuracy choice, not a formality, since BCa works farther into the
# tails of the bootstrap distribution than the percentile interval
```

```
# does (see Details).
reliability_alpha(data = items, ci_method = "bca", B = 500, seed = 113)

# Full information maximum likelihood with an auxiliary variable.
# Missingness on y2 depends on an auxiliary z (missing at random given
# z), so listwise deletion is biased and FIML with z is not. Supplying
# aux implies missing = "fiml"; the N_complete row shows how many
# rows listwise deletion would have kept.
z <- eta + rnorm(200, sd = 0.5)
d <- data.frame(items, z = z)
d$y2[runif(200) < plogis(-1 + 1.5 * as.numeric(scale(z)))] <- NA
reliability_alpha(data = d, aux = "z")
```

`reliability_H` *Maximal Reliability Coefficient* H *(Hancock & Mueller, 2001)*

Description

Computes the Hancock-Mueller (2001) maximal-reliability coefficient *H* from a vector of standardized factor loadings; *H* is the reliability of the optimally-weighted composite of a set of indicators of a single latent construct. It is uniformly greater than or equal to coefficient alpha and McDonald's omega for the same data, so it sets a useful upper bound on what reliability can plausibly be for that indicator set. A delta method confidence interval is reported when standard errors of the standardized loadings are supplied.

Usage

```
reliability_H(loadings, se_loadings = NULL, conf_level = 0.95)
```

Arguments

`loadings`	Numeric vector of standardized factor loadings, each in $(-1, 1)$. At least 2 loadings are required.
`se_loadings`	Optional vector of standard errors of the standardized loadings (same length as `loadings`). When supplied, a delta method CI on *H* is reported.
`conf_level`	Confidence level for the CI. Default `0.95`.

Details

Definition. For p indicators of a single latent factor with standardized loadings $\lambda_1, \ldots, \lambda_p$, Hancock & Mueller (2001) showed that the maximum reliability achievable by any linear composite of the indicators is

$$H \;=\; \frac{\sum_{i=1}^{p} \lambda_i^2/(1-\lambda_i^2)}{1 + \sum_{i=1}^{p} \lambda_i^2/(1-\lambda_i^2)}.$$

Equivalently, defining $\theta_i = \lambda_i^2/(1-\lambda_i^2)$ (the signal-to-noise ratio for indicator i), $H = \sum \theta_i/(1 + \sum \theta_i)$. As p grows or as the individual loadings grow toward 1, $H \to 1$.

Relationship to coefficient alpha and omega. Coefficient alpha (`reliability_alpha`) is the reliability of the *equally-weighted* sum of indicators; *H* is the reliability of the *optimally-weighted* composite. Hancock & Mueller (2001) prove $H \geq \omega \geq \alpha$ for a unidimensional indicator set, with equality only when all loadings are equal. *H* is therefore most useful for diagnostics: if *H* is much higher than alpha, the standard composite is leaving reliability on the table.

Confidence interval via the delta method. Conditional on standard errors $\mathrm{SE}(\hat{\lambda}_i)$, the delta method variance of *H* is

$$\mathrm{Var}(\hat{H}) \approx \sum_{i=1}^{p} \left(\frac{\partial H}{\partial \lambda_i} \right)^2 \mathrm{SE}(\hat{\lambda}_i)^2,$$

with $\partial H / \partial \lambda_i = 2\lambda_i / [(1-\lambda_i^2)^2 (1+\sum_j \theta_j)^2]$. The CI is built on the $\mathrm{logit}(H)$ scale (mapping $[0,1]$ to the real line) and back-transformed, as recommended by Browne (1968) for bounded reliability coefficients.

Value

A `data.frame` with rows for the point estimate `reliability_H` and (when SEs are supplied) the lower / upper CI bounds and the delta method variance.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Browne, M. W. (1968). A comparison of factor analytic techniques. *Psychometrika, 33*(3), 267–334.

Hancock, G. R., & Mueller, R. O. (2001). Rethinking construct reliability within latent variable systems. In R. Cudeck, S. du Toit, & D. Sörbom (Eds.), *Structural equation modeling: Present and future* (pp. 195–216). Scientific Software International.

Kelley, K., & Cheng, Y. (2012). Estimation of and confidence interval formation for reliability coefficients of homogeneous measurement instruments. *Methodology, 8*, 39–50. doi:10.1027/1614-2241/a000036

Kelley, K., & Pornprasertmanit, S. (2016). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for composite measures. *Psychological Methods, 21*, 69–92. doi:10.1037/a0040086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Raykov, T. (1997). Estimation of composite reliability for congeneric measures. *Applied Psychological Measurement, 21*(2), 173–184. doi:10.1177/01466216970212006

Terry, L. J., & Kelley, K. (2012). Sample size planning for composite reliability coefficients: Accuracy in parameter estimation via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 371–401. doi:10.1111/j.20448317.2011.02030.x

See Also

reliability_alpha, reliability_omega, reliability

Other reliability: cohen_kappa(), diagnosis_agreement, fleiss_kappa(), icc(), reliability(), reliability_alpha(), reliability_kr20(), reliability_omega(), reliability_omega_categorical()

Examples

```
# 1. Five indicators with standardized loadings 0.6, 0.7, ..., 0.8:
reliability_H(loadings = c(0.6, 0.65, 0.70, 0.75, 0.80))

# 2. With per-loading standard errors from a CFA output:
reliability_H(loadings    = c(0.6, 0.7, 0.8),
              se_loadings = c(0.05, 0.04, 0.03))
```

---

reliability_kr20 *Kuder-Richardson Formula 20 (KR-20) With a Confidence Interval*

Description

Estimates Kuder-Richardson formula 20 (Kuder & Richardson, 1937) for a homogeneous composite scored on dichotomous (0/1) items, and returns a confidence interval for the population coefficient.

Usage

```
reliability_kr20(
  data,
 ci_method = c("feldt", "bonett", "fisher", "hakstian_whalen", "ml", "ml_logistic",
  "adf", "adf_logistic", "bootstrap_se", "bootstrap_se_logistic", "percentile", "bca",
    "none"),
  conf_level = 0.95,
  B = 10000,
  seed = NULL
)
```

Arguments

data	A numeric matrix or data frame of 0/1 item scores (rows are respondents, columns are items). Rows with any missing values are listwise-deleted. Non-binary values trigger an error.
ci_method	Method for constructing the confidence interval; see reliability_alpha for the full list. Defaults to "feldt".
conf_level	Confidence level for the interval (1 - Type I error rate). Defaults to 0.95.
B	Number of bootstrap replications when a bootstrap method is selected. Defaults to 10000.
seed	Random number seed used for bootstrap reproducibility. Defaults to NULL, which leaves the user's current RNG state intact; supply an integer for reproducibility.

Details

Kuder and Richardson's (1937) formula 20 for a J-item composite of binary items is

$$KR_{20} = \frac{J}{J-1}\left(1 - \frac{\sum_j p_j q_j}{s_Y^2}\right),$$

where p_j is the proportion of respondents endorsing item j (i.e., scoring 1), $q_j = 1 - p_j$, and s_Y^2 is the variance of the composite score. For dichotomous items $p_j q_j$ is the item variance, so KR-20 is algebraically identical to coefficient α (Guttman, 1945; Cronbach, 1951) computed from the item covariance matrix. KR-20 predates coefficient α by 14 years and Feldt's (1965) F-distribution interval was derived specifically for the sampling distribution of KR-20.

Because KR-20 is a special case of coefficient α (Guttman, 1945; Cronbach, 1951), the same considerations apply: KR-20 equals the population reliability of the composite under essential τ-equivalence (i.e., equal factor loadings) and serves as a lower bound otherwise. For dichotomous items see also `reliability_omega_categorical` (categorical omega), which treats the items via a probit-link single-factor model and does not assume equal loadings.

Available confidence interval methods (set via `ci_method`) are the same as for `reliability_alpha`; see that function's *Details* for full descriptions. The default for KR-20 is `"feldt"`, the F-distribution interval originally developed for KR-20.

The menu also includes the nonparametric bootstrap intervals `"percentile"`, `"bca"`, `"bootstrap_se"`, and `"bootstrap_se_logistic"`, which resample the rows of `data` with replacement `B` times and recompute KR-20 on each replication (Efron & Tibshirani, 1993). They are worth the cost when the closed forms are least trustworthy, which for dichotomous items means highly unbalanced item difficulties or a sample size too small for the normal-theory derivations behind `"feldt"` and `"bonett"`. No bootstrap runs unless `ci_method` asks for one; when it does, the default is `B = 10000` replications, and supplying `seed` makes the interval reproducible. Replications on which KR-20 cannot be computed are set aside and counted, and the interval is computed from the replications that return a value; the counts travel in the `bootstrap` attribute described under `reliability_alpha`, print under the table, and are stated in a warning whenever a replication was set aside.

Comparison with other packages. The **psych** package computes the same quantity via `alpha` (since α on 0/1 data *is* KR-20). `reliability_kr20` restricts input to raw 0/1 data so the dichotomous-items assumption cannot be quietly violated, presents the historical formula in the documentation, and accompanies the point estimate with a confidence interval drawn from the methods compared in Kelley and Pornprasertmanit (2016).

Value

A `data.frame` with columns `term` and `value` and rows `"estimate"` (sample KR-20), `"se"` (standard error on the coefficient scale, `NA` for methods that do not produce one; for the transformation-based intervals `"fisher"`, `"bonett"`, and `"hakstian_whalen"` it is the delta method back-transform of the transformation-scale standard error), `"se_transformed"` (only for those transformation-based intervals: the standard error on the transformation scale, with the scale named in the attribute `se_transform_scale`: `"fisher_z"`, `"log(1-alpha)"`, or `"cube_root"`), `"lower_limit"` and `"upper_limit"` (clamped to [0, 1]), `"conf_level"`, `"N"` (effective sample size after listwise deletion), `"N_complete"` (the complete cases; equal to `"N"` here, and carried so the whole reliability family returns one shape), and `"J"` (number of items). Attributes `coefficient` (`"kr20"`) and `ci_method` record the computation; bootstrap calls also record `B` and the replication counts in the `bootstrap` attribute (see `reliability_alpha`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika, 16*(3), 297–334.

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, NY: Chapman & Hall/CRC.

Feldt, L. S. (1965). The approximate sampling distribution of Kuder-Richardson reliability coefficient twenty. *Psychometrika, 30*, 357–370.

Guttman, L. (1945). A basis for analyzing test-retest reliability. *Psychometrika, 10*(4), 255–282.

Kelley, K., & Cheng, Y. (2012). Estimation of and confidence interval formation for reliability coefficients of homogeneous measurement instruments. *Methodology, 8*, 39–50. doi:10.1027/1614-2241/a000036

Kelley, K., & Pornprasertmanit, S. (2016). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for composite measures. *Psychological Methods, 21*, 69–92. doi:10.1037/a0040086

Kuder, G. F., & Richardson, M. W. (1937). The theory of the estimation of test reliability. *Psychometrika, 2*, 151–160.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Terry, L. J., & Kelley, K. (2012). Sample size planning for composite reliability coefficients: Accuracy in parameter estimation via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 371–401. doi:10.1111/j.20448317.2011.02030.x

See Also

`reliability` (general wrapper), `reliability_alpha`, `reliability_omega_categorical`, `alpha`.

Other reliability: `cohen_kappa()`, `diagnosis_agreement`, `fleiss_kappa()`, `icc()`, `reliability()`, `reliability_H()`, `reliability_alpha()`, `reliability_omega()`, `reliability_omega_categorical()`

Examples

```
set.seed(113)
# Ten dichotomous items with a single underlying ability.
N <- 300
J <- 10
ability <- rnorm(N)
loadings <- rep(0.6, J)
latent <- outer(ability, loadings) +
          matrix(rnorm(N * J, sd = sqrt(1 - 0.6^2)), N, J)
items <- (latent > 0) * 1
colnames(items) <- paste0("y", seq_len(J))

reliability_kr20(data = items)
reliability_kr20(data = items, ci_method = "bonett")
```

```
# A bootstrap interval recomputes KR-20 on each of B resamples of the
# rows. B = 500 keeps the example quick; a reported interval deserves
# the default B = 10000, and seed makes the interval reproducible.
reliability_kr20(data = items, ci_method = "percentile", B = 500,
                 seed = 113)
```

`reliability_omega` *Coefficient Omega (McDonald) With a Confidence Interval*

Description

Estimates McDonald's (1999) coefficient ω for a homogeneous composite score from a single-factor confirmatory factor analysis model and returns a confidence interval for the population coefficient. The `denominator` argument selects whether the total variance in the denominator of ω is estimated directly from the data (`"observed"`, the default: robust omega) or taken from the fitted model (`"model_implied"`); see *Details* for the properties of each choice. A bootstrap confidence interval is never run unless requested; see `ci_method`.

Usage

```
reliability_omega(
  data = NULL,
  S = NULL,
  N = NULL,
  ci_method = c("mlr", "ml", "mlr_logistic", "ml_logistic", "likelihood", "adf",
   "adf_logistic", "feldt", "fisher", "bonett", "hakstian_whalen", "bootstrap_se",
    "bootstrap_se_logistic", "percentile", "bca", "none"),
  denominator = c("observed", "model_implied"),
  missing = c("listwise", "fiml"),
  aux = NULL,
  conf_level = 0.95,
  B = 10000,
  seed = NULL
)
```

Arguments

`data`	A numeric matrix or data frame of item scores (rows are respondents, columns are items). Either `data` or `S` must be supplied. How incomplete rows are handled is governed by `missing`: listwise deletion by default, or full information maximum likelihood with `missing = "fiml"`. When `aux` is supplied, `data` contains both the item columns and the auxiliary columns, and the columns not named in `aux` are the items.

S	A symmetric covariance matrix among the items. If supplied, N must also be supplied; methods that require raw data ("mlr*", "adf*", "bootstrap_*", "percentile", "bca") are then unavailable, as are missing = "fiml" and aux, since a covariance matrix has no incomplete cases.
N	Total sample size; required when S is supplied.
ci_method	Method for constructing the confidence interval. See *Details*. When not supplied: for denominator = "model_implied" the default is "mlr" with raw data (use "ml" with covariance input); for denominator = "observed" (robust omega) no interval is computed by default, because its interval is bootstrap based and a bootstrap is never run unless requested. Ask for "percentile" or "bca" to obtain the recommended interval. The closed forms, the Wald intervals, and the profile likelihood apply to denominator = "model_implied" only; a combination that does not apply stops with a message naming the denominator that does (see *Details*).
denominator	How the total variance in the denominator of ω is estimated: "observed" (default; robust omega) uses the variance of the composite estimated directly from the data; "model_implied" uses the total variance reproduced by the fitted single-factor model. See *Details*. With "observed", ci_method must be a bootstrap method or "none".
missing	How incomplete rows of data are handled: "listwise" (the default; complete-case analysis, the historical behavior) or "fiml" (full information maximum likelihood, using every case with at least one observed item). See *Details*.
aux	Optional character vector naming auxiliary variable columns of data, entered as saturated correlates under full information maximum likelihood. Supplying aux implies missing = "fiml". See *Details*.
conf_level	Confidence level for the interval. Defaults to 0.95.
B	Number of bootstrap replications when a bootstrap method is selected. Defaults to 10000.
seed	Random number seed used for bootstrap reproducibility. Defaults to NULL, which leaves the user's current RNG state intact; supply an integer for reproducibility.

Details

Coefficient ω is a population coefficient of determination. Write the composite $Y = \sum_j X_j$ through its measurement decomposition, $Y = E[Y \mid \eta] + e$, the regression of the observed composite on the latent variable η it is intended to measure. By the law of total variance, the proportion of composite variance attributable to η is $\mathrm{Var}(E[Y \mid \eta])/\mathrm{Var}(Y)$ (McDonald, 1999, 2011). Under the congeneric single-factor model, in which item j has factor loading λ_j and error variance ψ_j^2, the numerator equals $\left(\sum_j \lambda_j\right)^2$ and the population coefficient is

$$\omega = \frac{\left(\sum_j \lambda_j\right)^2}{\sigma_Y^2},$$

with σ_Y^2 the variance of the composite. Coefficient ω relaxes the equal-loadings assumption underlying coefficient α (Guttman, 1945; Cronbach, 1951), so on congeneric scales ω estimates population reliability whereas α underestimates it.

The denominator argument. The sample estimate substitutes **lavaan** estimates of the loadings into the numerator; the two `denominator` settings differ in how σ_Y^2 is estimated.

`"observed"` **(default)** The variance of the composite estimated directly from the data (the sum of all elements of the unrestricted covariance matrix, on the same maximum likelihood divisor as the fitted loadings). This is the coefficient that Kelley and Pornprasertmanit (2016) call hierarchical omega (`MBESS::ci.reliability(type = "hierarchical")`); the DMAR documentation refers to it as *robust omega*, and it is the default because its denominator estimates the variance of the composite consistently whether or not the single-factor model is correctly specified.

`"model_implied"` The total variance reproduced by the fitted single-factor model, $\left(\sum_j \hat{\lambda}_j\right)^2 + \sum_j \hat{\psi}_j^2$. This is the textbook form of $\hat{\omega}$, correct exactly when the one-factor model reproduces the composite variance.

The choice matters only insofar as the single-factor model is misspecified, and the properties of each setting can be stated exactly.

- If the single-factor model is correctly specified, the two definitions coincide in the population, and both estimators converge to the same value. Nothing is given up, in that sense, by either choice.
- The observed total variance is a consistent estimator of $\mathrm{Var}(Y)$ whether or not the single-factor model is correctly specified; the model implied total variance is consistent for $\mathrm{Var}(Y)$ only when the model is correct.
- Under misspecification (for example, a minor unmodeled factor or correlated errors), only `"observed"` retains the coefficient of determination interpretation: the proportion of the variance of the composite users actually compute that is attributable to the fitted common factor. With `"model_implied"` the estimate becomes a ratio of two model derived quantities whose denominator is no longer the variance of any composite a user scores.
- Under a correctly specified model with normal items, the model implied denominator uses the model structure and can be a slightly more efficient estimator of σ_Y^2 in finite samples. In the simulations of Kelley and Pornprasertmanit (2016) the practical differences between the two coefficients were negligible when the model held, while interval coverage under modest model error favored the observed denominator paired with a bootstrap interval.

Those simulation results are why Kelley and Pornprasertmanit (2016) recommend the observed denominator with a bootstrap confidence interval when unidimensionality is only approximate, which is common with real items.

Two cautions frame the choice. First, no denominator repairs a misspecified measurement model: the fitted loadings absorb part of whatever structure the single-factor model omits, so the numerator is affected under either setting. Assess the single-factor model (for example with `cfa_1`) before interpreting any ω variant, and model real multidimensionality directly rather than patching over it. Second, the naming history is worth knowing. The observed denominator coefficient was introduced as hierarchical omega (Kelley & Pornprasertmanit, 2016; **MBESS** type `"hierarchical"`), a name motivated by a hierarchical factor logic: model misfit is viewed as a set of minor common factors (visible as residual correlations), the single factor is retained as an approximation, and the coefficient isolates the variance attributable to the general factor alone, expressed relative to the observed variance of the unweighted composite. It is not the bifactor coefficient ω_H of Zinbarg,

Revelle, Yovel, and Li (2005), whose numerator comes from the general factor loadings of an explicitly multidimensional model. Upon reflection, the authors would have named the coefficient for its behavior rather than for the hierarchical motivation, as observed omega or robust omega; **DMAR** uses *robust omega*, with the qualifications that word requires. The robustness is to misspecification of the *total variance* only, since the numerator remains model based under either denominator; it is not the outlier robustness of Zhang and Yuan (2016), and it is separate from the robust maximum likelihood standard errors available through `ci_method`. Robust omega also shares a design principle with categorical omega: in both, the total variance in the denominator is not taken from the fitted factor model. The reliability vignette develops this framing.

Available confidence interval methods (set via `ci_method`):

`"ml"`, `"ml_logistic"` Wald interval using the maximum likelihood standard error from **lavaan** (Raykov, 2002). The `_logistic` variant applies Browne's (1982) logit transformation.

`"mlr"`, `"mlr_logistic"` Wald interval using the robust (Satorra & Bentler, 1994) standard error. Default; recommended among closed-form methods when item distributions deviate from normality (Kelley & Pornprasertmanit, 2016).

`"likelihood"` Profile likelihood interval: the set of population values not rejected by the likelihood ratio test, located by refitting the model under the nonlinear constraint that the model implied reliability equals each candidate value. It respects the [0, 1] range and is not forced to be symmetric about the estimate. Maximum likelihood; available with raw data or covariance input, for `denominator = "model_implied"` only.

`"adf"`, `"adf_logistic"` Wald interval using weighted least squares ("ADF") estimation (Browne, 1984). Requires raw data and a relatively large sample size.

`"feldt"`, `"fisher"`, `"bonett"`, `"hakstian_whalen"` Closed-form intervals derived for coefficient α. They apply mechanically to ω but their coverage performance for ω is generally inferior to the maximum likelihood and bootstrap intervals (Kelley & Pornprasertmanit, 2016).

`"bootstrap_se"`, `"bootstrap_se_logistic"`, `"percentile"`, `"bca"` Nonparametric bootstrap intervals (Efron & Tibshirani, 1993): the rows of `data` are resampled with replacement `B` times and ω is recomputed, with the factor model refit, on each replication. `"percentile"` takes the interval limits from the empirical quantiles of the bootstrap estimates; it respects [0, 1] and is not forced to be symmetric about the estimate. `"bca"` (bias-corrected and accelerated) adjusts the two quantile positions for median bias, estimated from the bootstrap distribution, and for the rate at which the estimator's variance changes with the parameter, the acceleration, estimated by the jackknife; those two adjustments make it second-order accurate where the percentile interval is first-order accurate (DiCiccio & Efron, 1996). The acceleration is estimated as **boot** estimates it, by regression on the resampling frequencies when at least as many replications as cases returned a value and by the jackknife otherwise; a leave-one-out refit that fails is counted and set aside rather than stopping the call (as one did through version 1.0.0), and when the acceleration or the bias correction cannot be estimated at all the percentile interval is returned in its place, with a warning and the substitution recorded in the `ci_method` attribute; see `reliability_omega_categorical` for the full account. `"bootstrap_se"` uses the standard deviation of the bootstrap estimates as a standard error in a normal-theory interval, built on the logit scale for the `_logistic` variant so the endpoints respect [0, 1]. Replications whose model refit does not converge are set aside, and the interval is computed from the replications that return a value, whose quantiles the percentile and BCa limits are; the counts, and why the rest were set aside, travel in the `bootstrap` attribute (see *Value*), print under the table, and are stated in a warning whenever a replication was set aside. The default `B = 10000`

is an accuracy choice: the BCa adjustment pushes the working quantiles farther into the tails of the bootstrap distribution than the percentile interval uses, and stabilizing them takes more replications than the customary 2000; reduce `B` for exploration, not for a reported analysis. Recommended when assumptions of parametric methods are questionable. Requires raw data and the **boot** package; supply `seed` for run-to-run reproducibility.

`"none"` Return only the point estimate.

Which intervals apply to which denominator.

`denominator = "model_implied"` Every method on the menu: the Wald intervals `"ml"`, `"mlr"`, and `"adf"` with their `_logistic` variants, the profile `"likelihood"` interval, the alpha-derived closed forms `"feldt"`, `"fisher"`, `"bonett"`, and `"hakstian_whalen"`, the four bootstrap methods, and `"none"`.

`denominator = "observed"` **(robust omega, the default)** The four bootstrap methods (`"percentile"`, `"bca"`, `"bootstrap_se"`, `"bootstrap_se_logistic"`) and `"none"` only. The delta method standard errors, the profile likelihood, and the alpha-derived closed forms are derived under the model implied ratio and do not account for sampling of the observed denominator, so `"mlr"`, the default interval of the model implied coefficient, is not an interval for robust omega.

A request for a combination that does not apply, such as `ci_method = "mlr"` with the default denominator, stops with a message naming the denominator that does; nothing falls back silently. Covariance input (`S`) and `missing = "fiml"` narrow the menu further, as the `S` argument and the paragraph on missing data below describe, and each stops the same way. The pairing of robust omega with the bootstrap is not a limitation in practice, since the bootstrap is the interval Kelley and Pornprasertmanit (2016) recommend for the observed denominator coefficient in any case. Because no analysis in **DMAR** runs a bootstrap unless the user requests one, the default for robust omega is the point estimate with no interval, accompanied by a message naming the call that produces the recommended interval; request `ci_method = "percentile"` or `"bca"` to obtain it.

Missing data and auxiliary variables. By default incomplete rows are listwise-deleted; `missing = "fiml"` keeps every case with at least one observed item and fits the single-factor model by full information maximum likelihood, which is consistent and efficient under the missing at random (MAR) assumption. The `aux` argument names auxiliary variables: columns of `data` that are not part of the composite but are correlated with the items or with the reasons values are missing. They enter as *saturated correlates* (Graham, 2003): correlated freely with each other and with every item's residual, never loading on the factor, so the measurement model is undisturbed while FIML uses their information; a good auxiliary also makes MAR itself more plausible (Collins, Schafer, & Kam, 2001). Supplying `aux` implies `missing = "fiml"`; combining it with an explicit `missing = "listwise"` is an error, and listwise deletion remains the default so no existing result changes. Under `missing = "fiml"` the denominators are estimated as follows: with `"model_implied"` the fitted FIML model supplies the total variance directly, and with `"observed"` the total variance is the sum of the FIML estimate of the item covariance matrix (from a saturated model over the items and any auxiliaries), which is the estimate of $\mathrm{Var}(Y)$ that uses the partially observed rows; it is already on the maximum likelihood divisor, matching the fitted loadings. The available intervals under `"fiml"` are `"ml"`, `"mlr"` (Yuan & Bentler, 2000, robust), their `_logistic` variants, and the bootstrap methods, which resample rows including the partially observed ones and refit by FIML; the complete-data closed forms, `"adf"`, and `"likelihood"` are errors rather than silent fallbacks to listwise deletion. Multiple imputation is a different feature with a different interface and is out of scope here.

Comparison with other packages. The **psych** package provides `omega`, which fits a Schmid-Leiman hierarchical factor model and reports several variants of ω (ω_t, ω_h) alongside extensive psychometric diagnostics. `reliability_omega` in **DMAR** differs in emphasis: it implements McDonald's ω from a single-factor (congeneric) model and accompanies the point estimate with a confidence interval drawn from the methods compared in Kelley and Pornprasertmanit (2016). The same denominator distinction appears in semTools' `compRelSEM()` as its `obs.var` argument.

Value

A `data.frame` with columns `term` and `value` and rows `"estimate"` (sample coefficient ω), `"se"` (standard error on the coefficient scale, `NA` for methods that do not produce one; for the transformation-based intervals `"fisher"`, `"bonett"`, and `"hakstian_whalen"` it is the delta method back-transform of the transformation-scale standard error), `"se_transformed"` (only for those transformation-based intervals: the standard error on the transformation scale, with the scale named in the attribute `se_transform_scale`: `"fisher_z"`, `"log(1-alpha)"`, or `"cube_root"`), `"lower_limit"` and `"upper_limit"` (clamped to [0, 1]), `"conf_level"`, `"N"` (the cases the analysis used: the complete cases under listwise deletion, every case with at least one observed item under `missing = "fiml"`), `"N_complete"` (the complete cases; equal to `"N"` under listwise deletion and for covariance input), and `"J"`. Attributes `coefficient` (`"omega"`), `ci_method`, `denominator`, `missing`, and (when supplied) `aux` record the computation; bootstrap calls also record `B`. A bootstrap call also carries the attribute `bootstrap`, a list recording the replications the interval conditions on: `attempted` (`B`), `returned` (the replications that returned a finite value; the standard error is their standard deviation and the percentile and BCa limits are their quantiles), `failed` (the difference), `reasons` (a named count of why replications were set aside: `error` when the refit raised an error, `non_finite` when it returned no finite value, as a fit that did not converge does), `error_messages` (the distinct error messages with their counts, empty when none), `acceleration` and `acceleration_method` (the BCa acceleration used and whether it came from the regression or the jackknife; `NA` unless a BCa interval was formed), and `jackknife` (the same `attempted`, `returned`, `failed`, `reasons`, and `error_messages` for the leave-one-out refits behind the acceleration, all zero when no jackknife ran). The printed table shows these counts in one line, and a warning states them whenever a replication was set aside.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bonett, D. G. (2002). Sample size requirements for testing and estimating coefficient alpha. *Journal of Educational and Behavioral Statistics, 27*(4), 335–340. doi:10.3102/10769986027004335

Browne, M. W. (1982). Covariance structures. In D. M. Hawkins (Ed.), *Topics in applied multivariate analysis* (pp. 72–141). Cambridge, UK: Cambridge University Press.

Browne, M. W. (1984). Asymptotically distribution-free methods for the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology, 37*, 62–83.

Satorra, A., & Bentler, P. M. (1994). Corrections to test statistics and standard errors in covariance structure analysis. In A. von Eye & C. C. Clogg (Eds.), *Latent variables analysis: Applications for developmental research* (pp. 399–419). Thousand Oaks, CA: Sage.

Yuan, K.-H., & Bentler, P. M. (2000). Three likelihood-based methods for mean and covariance structure analysis with nonnormal missing data. *Sociological Methodology, 30*, 165–200.

Collins, L. M., Schafer, J. L., & Kam, C.-M. (2001). A comparison of inclusive and restrictive strategies in modern missing data procedures. *Psychological Methods, 6*, 330–351. doi:10.1037/1082989X.6.4.330

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika, 16*(3), 297–334.

DiCiccio, T. J., & Efron, B. (1996). Bootstrap confidence intervals. *Statistical Science, 11*(3), 189–228.

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, NY: Chapman & Hall/CRC.

Feldt, L. S. (1965). The approximate sampling distribution of Kuder-Richardson reliability coefficient twenty. *Psychometrika, 30*, 357–370.

Graham, J. W. (2003). Adding missing-data-relevant variables to FIML-based structural equation models. *Structural Equation Modeling, 10*(1), 80–100. doi:10.1207/S15328007SEM1001_4

Guttman, L. (1945). A basis for analyzing test-retest reliability. *Psychometrika, 10*(4), 255–282.

Hakstian, A. R., & Whalen, T. E. (1976). A *k*-sample significance test for independent alpha coefficients. *Psychometrika, 41*, 219–231.

Kelley, K. (2007a). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K. (2007b). Methods for the behavioral, educational, and social sciences: An R package. *Behavior Research Methods, 39*(4), 979–984. doi:10.3758/BF03192993

Kelley, K., & Cheng, Y. (2012). Estimation of and confidence interval formation for reliability coefficients of homogeneous measurement instruments. *Methodology, 8*, 39–50. doi:10.1027/1614-2241/a000036

Kelley, K., & Pornprasertmanit, S. (2016). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for composite measures. *Psychological Methods, 21*, 69–92. doi:10.1037/a0040086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

McDonald, R. P. (1999). *Test theory: A unified treatment*. Mahwah, NJ: Lawrence Erlbaum Associates.

McDonald, R. P. (2011). Measuring latent quantities. *Psychometrika, 76*, 511–536. doi:10.1007/s1133601192237

Raykov, T. (2002). Analytic estimation of standard error and confidence interval for scale reliability. *Multivariate Behavioral Research, 37*, 89–103. doi:10.1207/S15327906MBR3701_04

Satorra, A., & Bentler, P. M. (2001). A scaled difference chi-square test statistic for moment structure analysis. *Psychometrika, 66*(4), 507–514. doi:10.1007/BF02296192

Terry, L. J., & Kelley, K. (2012). Sample size planning for composite reliability coefficients: Accuracy in parameter estimation via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 371–401. doi:10.1111/j.20448317.2011.02030.x

Zhang, Z., & Yuan, K.-H. (2016). Robust coefficients alpha and omega and confidence intervals with outlying observations and missing data: Methods and software. *Educational and Psychological Measurement, 76*, 387–411. doi:10.1177/0013164415594658

Zinbarg, R. E., Revelle, W., Yovel, I., & Li, W. (2005). Cronbach's α, Revelle's β, and McDonald's ω_H: Their relations with each other and two alternative conceptualizations of reliability. *Psychometrika, 70*, 123–133. doi:10.1007/s1133600309747

See Also

reliability (general wrapper), reliability_omega_categorical (categorical omega for ordered items), reliability_alpha, cfa_1 (single-factor CFA used internally), omega.

Other reliability: cohen_kappa(), diagnosis_agreement, fleiss_kappa(), icc(), reliability(), reliability_H(), reliability_alpha(), reliability_kr20(), reliability_omega_categorical()

Examples

```
set.seed(113)
J <- 6
loadings <- seq(0.5, 0.8, length.out = J)
eta <- rnorm(200)
errors <- matrix(rnorm(200 * J), 200, J) %*% diag(sqrt(1 - loadings^2))
items <- sweep(matrix(rep(eta, J), 200, J), 2, loadings, `*`) + errors
colnames(items) <- paste0("y", seq_len(J))

# Default: robust omega, point estimate only (no bootstrap is run
# unless requested; a message names the call that produces the
# recommended interval).
reliability_omega(data = items)

# The closed-form standard errors are derived under the model implied
# ratio, so with the observed denominator the interval comes from a
# bootstrap, which refits the single factor model once per
# replication. B = 40 keeps the example quick, and it is about the
# smallest count from which a 95 percent percentile interval can be
# read at all; a reported interval deserves the default B = 10000,
# an accuracy choice rather than a formality (see Details). The
# percentile interval is what Kelley and Pornprasertmanit (2016)
# recommend for this coefficient, with ci_method = "bca" as the
# alternative, and seed makes the interval reproducible.
reliability_omega(data = items, ci_method = "percentile", B = 40,
                  seed = 113)

# Model implied denominator with its closed-form robust ML interval.
reliability_omega(data = items, denominator = "model_implied")

# Two further routes into the same coefficient. The first works from
# the summary statistics a paper reports, a covariance matrix and its
# sample size, with the maximum likelihood interval that summary
# input supports.
reliability_omega(S = cov(items), N = 200,
                  denominator = "model_implied", ci_method = "ml")
```

```
# The second is full information maximum likelihood with an auxiliary
# variable, where missingness on y2 depends on an auxiliary z (missing
# at random given z). Supplying aux implies missing = "fiml"; the
# N_complete row shows how many rows listwise deletion would have
# kept.
z <- eta + rnorm(200, sd = 0.5)
d <- data.frame(items, z = z)
d$y2[runif(200) < plogis(-1 + 1.5 * as.numeric(scale(z)))] <- NA
reliability_omega(data = d, aux = "z", denominator = "model_implied")
```

`reliability_omega_categorical` *Categorical Omega for Ordered-Categorical Items, With a Confidence Interval*

Description

Estimates categorical omega (ω_C; Green & Yang, 2009; Kelley & Pornprasertmanit, 2016) for a homogeneous composite of ordered-categorical items and returns a bootstrap confidence interval.

Usage

```
reliability_omega_categorical(
  data,
 ci_method = c("bca", "percentile", "bootstrap_se", "bootstrap_se_logistic", "none"),
  conf_level = 0.95,
  B = 10000,
  seed = NULL
)
```

Arguments

`data`	A numeric matrix or data frame of ordered-categorical item scores (integer codes for the categories). Rows with any missing values are listwise-deleted.
`ci_method`	Method for constructing the confidence interval. See *Details*. When not supplied, no interval is computed: every interval for categorical omega is bootstrap based, and a bootstrap is never run unless requested. Ask for `"bca"` (the recommended method) or `"percentile"` to obtain one.
`conf_level`	Confidence level for the interval. Defaults to `0.95`.
`B`	Number of bootstrap replications. Defaults to `10000`.
`seed`	Random number seed used for bootstrap reproducibility. Defaults to `NULL`, which leaves the user's current RNG state intact; supply an integer for reproducibility.

Details

Categorical omega is designed for items measured on an ordered categorical scale (e.g., Likert items). It uses a probit-link single-factor model in which each observed item X_j is modeled as a categorization of an underlying continuous response variable X_j^* via thresholds $t_{j,c}$ (Muthén, 1984; Millsap & Yun-Tein, 2004), fit by diagonally weighted least squares with mean- and variance-adjusted test statistic (WLSMV), the standard estimator for ordered categorical items. With the delta parameterization ($Var(X_j^*) = 1$), the population categorical omega is

$$\omega_C = \frac{\sum_{j=1}^{J}\sum_{j'=1}^{J}\sigma_{jj'}(\lambda_j\lambda_{j'})}{\sum_{j=1}^{J}\sum_{j'=1}^{J}\sigma_{jj'}\left(\rho_{X_j^*X_{j'}^*}\right)},$$

where $\sigma_{jj'}(r)$ is the model implied covariance of $(X_j, X_{j'})$ computed from a bivariate normal CDF over pairs of category thresholds and a correlation r (Green & Yang, 2009, Eq. 13–14, with the full coefficient their Eq. 21; Kelley & Pornprasertmanit, 2016, Eq. 17–18). The numerator uses model implied polychoric correlations ($\lambda_j\lambda_{j'}$), while the denominator uses observed polychoric correlations estimated from the data via a saturated bivariate model.

Kelley and Pornprasertmanit (2016) found in extensive Monte Carlo simulation that the bias-corrected and accelerated (BCa) bootstrap confidence interval for categorical omega achieved acceptable coverage across a wide variety of threshold patterns, sample sizes, item counts, and population reliability values, with one documented exception: coverage dipped somewhat below the acceptable range when the number of items and the population reliability were both high. They specifically recommend BCa for categorical omega. Because no bootstrap runs in **DMAR** unless the user requests one, the default output is the point estimate with a message naming the call that produces the recommended interval; request `ci_method = "bca"` to obtain it. The interval is not a quick one. Every replication refits the ordered-categorical factor model and the saturated polychoric model behind the denominator, a fraction of a second for a scale of moderate length, so the recommended interval at the default `B = 10000` runs for tens of minutes; when `B` is below the number of cases, the jackknife behind the BCa acceleration adds one such refit per case (at the default `B` the acceleration is read from the replicates themselves at no extra cost; see the `"bca"` entry below). The example on this page therefore stops at the point estimate; the reported analysis adds `ci_method = "bca"` to the same call, with `seed` for reproducibility.

When to use. Use `reliability_omega_categorical` when items are ordered-categorical, especially when (a) the number of categories is small (e.g., two to five), (b) item distributions are skewed, or (c) threshold patterns differ markedly across items. In Kelley and Pornprasertmanit's (2016) Study 3, treating ordered items as continuous and using `reliability_omega` with the observed total variance in the denominator achieved acceptable coverage only when threshold patterns were similar across items; in their experience that condition is rare in practice.

Available confidence interval methods (set via `ci_method`). Every interval here is bootstrap based: the rows of `data` are resampled with replacement `B` times (10000 by default) and categorical omega is recomputed, with the full WLSMV model refit, on each replication (Efron & Tibshirani, 1993). Replications whose refit fails or does not converge, most common with small samples and sparse response categories, are set aside, and the interval is computed from the replications that return a value: the standard error is their standard deviation, and the percentile and BCa limits are quantiles of the returned replicates, not of all `B`. How many were attempted, how many returned a value, and why the rest were set aside is recorded in the `bootstrap` attribute of the result (see *Value*), printed in one line under the table, and stated in a warning whenever a replication was set aside, so an

interval that conditions on a subset of the replications says so and by how much. Bootstrap results vary from run to run; supply `seed` for reproducibility.

`"bca"` The bias-corrected and accelerated bootstrap, the default and the specific recommendation of Kelley and Pornprasertmanit (2016). Where the percentile interval reads its limits directly off the empirical quantiles of the bootstrap estimates, BCa adjusts the two quantile positions for median bias (estimated from the bootstrap distribution) and for the rate at which the estimator's variance changes with the parameter (the acceleration, estimated by the jackknife), making it second-order accurate where the percentile interval is first-order accurate (DiCiccio & Efron, 1996). The adjusted quantile positions sit farther into the tails than the percentile interval uses, which is why the default `B = 10000` is larger than the customary 2000; reduce `B` for exploration, not for a reported analysis. The acceleration is estimated as **boot** estimates it: by regressing the bootstrap replicates on the resampling frequencies when at least as many replications as cases returned a value, at no extra cost, and otherwise by the jackknife, one leave-one-out refit per case. A leave-one-out refit that fails (when a case is the only respondent in one of an item's categories, for example) is counted and set aside, and the acceleration is estimated from the refits that returned a value; through version 1.0.0 one such failure made the acceleration `NA` and stopped the call with "estimated adjustment 'a' is NA". When the acceleration cannot be estimated at all (fewer than two leave-one-out refits returned a value), or the bias correction cannot be (every returned replicate on one side of the estimate), the percentile interval is returned in its place with a warning, and the `ci_method` attribute records the substitution as `"percentile (BCa acceleration could not be estimated: ...)"`. The counts of leave-one-out refits and the acceleration used are in the `bootstrap` attribute.

`"percentile"` Percentile bootstrap: the interval limits are the empirical quantiles of the bootstrap estimates.

`"bootstrap_se"`, `"bootstrap_se_logistic"` Wald intervals using the bootstrap standard deviation as a standard error, built on the logit scale for the `_logistic` variant so the endpoints respect [0, 1].

`"none"` Return only the point estimate.

Comparison with other packages. The **psych** package's `omega` fits a Schmid-Leiman hierarchical factor model on continuous (or treated-as-continuous) items and does not implement categorical omega in the sense of Green and Yang (2009). For ordered-categorical items `reliability_omega_categorical` is the appropriate choice; `polychoric` provides polychoric correlation estimation as a separate tool.

Value

A `data.frame` with columns `term` and `value` and rows `"estimate"` (sample ω_C), `"se"` (the bootstrap standard deviation across replications, already on the coefficient scale; `NA` for `ci_method = "none"`), `"lower_limit"` and `"upper_limit"` (clamped to [0, 1]), `"conf_level"`, `"N"`, `"N_complete"` (the complete cases; equal to `"N"` here, and carried so the whole reliability family returns one shape), and `"J"`. Attributes `coefficient` (`"omega_categorical"`), `ci_method`, and `B` record the computation. A bootstrap call also carries the attribute `bootstrap`, a list recording the replications the interval conditions on: `attempted` (`B`), `returned` (the replications that returned a finite value; the standard error is their standard deviation and the percentile and BCa limits are their quantiles), `failed` (the difference), `reasons` (a named count of why replications were set aside: `error` when the refit raised an error, `non_finite` when it returned no finite value, as a fit that did not converge does), `error_messages` (the distinct error messages with their counts, empty when none), `acceleration` and `acceleration_method` (the BCa acceleration used and whether it came from

the regression or the jackknife; `NA` unless a BCa interval was formed), and `jackknife` (the same `attempted`, `returned`, `failed`, `reasons`, and `error_messages` for the leave-one-out refits behind the acceleration, all zero when no jackknife ran). The printed table shows these counts in one line, and a warning states them whenever a replication was set aside.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

DiCiccio, T. J., & Efron, B. (1996). Bootstrap confidence intervals. *Statistical Science, 11*(3), 189–228.

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, NY: Chapman & Hall/CRC.

Green, S. B., & Yang, Y. (2009). Reliability of summed item scores using structural equation modeling: An alternative to coefficient alpha. *Psychometrika, 74*, 155–167. doi:10.1007/s11336-00890993

Kelley, K., & Cheng, Y. (2012). Estimation of and confidence interval formation for reliability coefficients of homogeneous measurement instruments. *Methodology, 8*, 39–50. doi:10.1027/1614-2241/a000036

Kelley, K., & Pornprasertmanit, S. (2016). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for composite measures. *Psychological Methods, 21*, 69–92. doi:10.1037/a0040086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Millsap, R. E., & Yun-Tein, J. (2004). Assessing factorial invariance in ordered-categorical measures. *Multivariate Behavioral Research, 39*(3), 479–515. doi:10.1207/s15327906mbr3903_4

Muthén, B. (1984). A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators. *Psychometrika, 49*(1), 115–132.

Terry, L. J., & Kelley, K. (2012). Sample size planning for composite reliability coefficients: Accuracy in parameter estimation via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 371–401. doi:10.1111/j.20448317.2011.02030.x

See Also

`reliability` (general wrapper), `reliability_omega` (use for continuous items), `reliability_kr20` (for dichotomous items), `omega`, `polychoric`.

Other reliability: `cohen_kappa()`, `diagnosis_agreement`, `fleiss_kappa()`, `icc()`, `reliability()`, `reliability_H()`, `reliability_alpha()`, `reliability_kr20()`, `reliability_omega()`

Examples

```
set.seed(113)
# Six 5-category items with a single latent factor.
N <- 200
J <- 6
```

```
loadings <- rep(0.7, J)
eta <- rnorm(N)
latent <- outer(eta, loadings) +
          matrix(rnorm(N * J), N, J) %*% diag(sqrt(1 - loadings^2))
items <- apply(latent, 2, function(x)
  as.integer(cut(x, breaks = c(-Inf, -1.5, -0.5, 0.5, 1.5, Inf),
                 labels = FALSE)))
colnames(items) <- paste0("y", seq_len(J))

# Default: point estimate only, with a message naming the call that
# produces the recommended interval.
reliability_omega_categorical(data = items)

# The same items treated as continuous, for contrast. With five
# categories and thresholds spread across the latent scale the two
# coefficients nearly agree on these data; the gap widens as the
# categories get coarser and as the thresholds move into the tails,
# which is where the categorical coefficient is worth its cost.
reliability_omega(data = items)
```

`responder_analysis` *Responder Analysis: Who Cleared the Threshold, by Group*

Description

The clinical and behavioral endpoint that mean differences hide: the proportion of each group whose outcome reaches a meaningful threshold (a minimal clinically important difference, a remission cut, a mastery criterion). For each group the function reports the responder count and proportion with a Wilson confidence interval; with exactly two groups it adds the risk difference with the Newcombe (1998) score-based hybrid interval and the number needed to treat; and across any number of groups it reports the omnibus chi square test of equal responder proportions. An optional `sweep` repeats the analysis over a grid of thresholds, making explicit how conclusions depend on where the line is drawn, disclosing the threshold dependence that any single-threshold claim leaves implicit.

Usage

```
responder_analysis(
  x,
  group,
  threshold,
  direction = c("ge", "le"),
  conf_level = 0.95,
  sweep = NULL
)
```

Arguments

x	Numeric vector of outcomes (for example, change scores).
group	Group labels, one per observation (coerced to factor; the first level is the reference for the two-group difference).
threshold	The cut defining response.
direction	"ge" (default): a responder has x >= threshold; "le": x <= threshold (for outcomes where lower is better).
conf_level	Confidence level for all intervals. Defaults to 0.95.
sweep	Optional numeric vector of additional thresholds; the analysis is repeated at each and stacked with a leading threshold column.

Details

Per-group intervals are Wilson score intervals (ci_proportion). The two-group risk difference uses Newcombe's method 10: the difference interval is assembled from the two Wilson limits, which keeps it inside [-1, 1] and well behaved at boundary counts. The number needed to treat is $1/|\Delta|$, with its interval from the inverted difference limits when the difference interval excludes zero; when it includes zero the NNT interval is reported as NA (the interval is disjoint and an interval on the NNT scale would mislead; Altman, 1998). Dichotomizing throws away information, so a responder analysis complements, never replaces, the analysis of the continuous outcome (Maxwell, Delaney, & Kelley, 2027).

Value

A tidy wide data.frame (class dmar_tbl). One row per group with group, n, responders, estimate (the proportion), lower_limit, upper_limit; with two groups, a difference row (second level minus first) and an nnt row; and a final omnibus row carrying chi_square, df, and p_value (columns that are NA on the other rows). When sweep is supplied, the same table is stacked per threshold with a leading threshold column.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Altman, D. G. (1998). Confidence intervals for the number needed to treat. *BMJ, 317*(7168), 1309–1312. doi:10.1136/bmj.317.7168.1309

Newcombe, R. G. (1998). Interval estimation for the difference between independent proportions: Comparison of eleven methods. *Statistics in Medicine, 17*(8), 873–890. doi:10.1002/(SICI)1097-0258(19980430)17:8<873::AIDSIM779>3.0.CO;2I

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

ci_proportion for the per-group interval; nnt_from_smd for the model-based route to the number needed to treat from a standardized mean difference; cliff_delta and proportion_of_superiority for dominance-style effect sizes on the continuous outcome.

Other effect size estimates: cles(), cliff_delta(), correction_for_attenuation(), eta_squared(), eta_squared_generalized(), eta_squared_partial(), expected_partial_r(), expected_r(), expected_smd(), nnt_from_smd(), omega_squared(), omega_squared_partial(), probability_of_superiority_pai proportion_of_superiority(), smd_trimmed()

Examples

```
# A two-arm trial: change scores, response defined as a gain of 10+.
set.seed(113)
change <- c(rnorm(60, 8, 9), rnorm(60, 13, 9))
arm    <- rep(c("control", "treatment"), each = 60)
responder_analysis(change, arm, threshold = 10)

# How threshold-dependent is that conclusion?
responder_analysis(change, arm, threshold = 10, sweep = c(5, 15))
```

---

sd_unbiased *Unbiased Estimate of the Population Standard Deviation Under Normality*

Description

Returns the unbiased estimate of the population standard deviation under normality. Although the sample variance s^2 is unbiased for σ^2 when computed with $N - 1$ in the denominator, its square root s is biased downward for σ because the square root function is concave (Jensen's inequality). sd_unbiased applies the classical Holtzman (1950) correction factor that exactly removes that bias under normality.

Usage

```
sd_unbiased(s = NULL, N = NULL, X = NULL)
```

Arguments

s	The usual estimate of the standard deviation (the square root of the unbiased variance s^2).
N	The sample size on which s is based.
X	Optional vector of raw scores from which s and N are inferred (s = sd(X), N = length(X)). Mutually exclusive with s and N.

Details

The sample variance computed with $N - 1$ in the denominator is unbiased for the population variance σ^2, but its square root s is biased downward for σ. Under normality, the multiplicative bias is

$$E[s] \;=\; \sigma \cdot c_N^{-1}, \qquad c_N \;=\; \sqrt{(N-1)/2}\cdot \Gamma((N-1)/2)\,/\,\Gamma(N/2),$$

(Holtzman, 1950). Multiplying s by c_N therefore yields an unbiased estimator of σ. The correction is non-trivial in small samples: c_N is about 1.064 at $N = 5$, 1.028 at $N = 10$, 1.009 at $N = 30$, and is essentially 1 by $N = 100$ (about 1.003). For most applied work the bias of s is small enough to ignore, but it matters when s feeds into downstream quantities (variance components, standardizers, planning calculations) at small N.

Implementation note: the factor c_N is computed via `lgamma` to avoid the overflow of `gamma` above $N \approx 340$, so the function is numerically stable for arbitrarily large N.

Value

A 1-row `data.frame` with columns `term` and `value`. The `term` value is `"sd"` and `value` is the unbiased estimate of σ.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Holtzman, W. H. (1950). The unbiased estimate of the population variance and standard deviation. *American Journal of Psychology*, *63*, 615–617.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

sd, cv

Examples

```
set.seed(113)
X <- rnorm(10, 100, 15)

# The plug-in estimate sqrt(s^2) is biased downward for sigma.
var(X)^.5

# Holtzman (1950) bias-corrected estimate, supplying s and N.
sd_unbiased(s = var(X)^.5, N = length(X))

# Equivalent call from the raw vector.
sd_unbiased(X = X)
```

`signal_to_noise_R2` *Signal to Noise Estimators for the Squared Multiple Correlation Coefficient*

Description

Computes five estimators of the population signal to noise ratio $\phi^2 = \rho^2/(1-\rho^2)$ associated with the squared multiple correlation coefficient ρ^2. Two are functions of the unadjusted and the Wherry-adjusted sample R^2; the other three are the Muirhead (1985) unique minimum variance unbiased estimators that improve substantially on the plug-in estimator at small N and modest numbers of predictors.

Usage

```
signal_to_noise_R2(R2, N, p)
```

Arguments

`R2`	The usual sample estimate of the squared multiple correlation coefficient (no degrees of freedom adjustment). Numeric scalar in $(0, 1)$.
`N`	Sample size.
`p`	Number of predictor variables.

Details

The signal to noise ratio $\phi^2 = \rho^2/(1-\rho^2)$ is a natural reparameterization of ρ^2 that is bounded only below (at zero) and so behaves more like a variance ratio than a proportion. It is also the noncentrality parameter (up to a factor of N) for the omnibus F-test of $\rho^2 = 0$ under fixed predictors; see `convert_R2_f`.

The five estimators returned, in increasing order of bias-correction machinery, are:

- `phi2_hat`: the plug-in estimator $\hat{\phi}^2 = R^2/(1-R^2)$. Biased upward in small samples because the sample R^2 is itself biased upward.
- `phi2_adj_hat`: the plug-in estimator applied to the Wherry-adjusted R^2. Removes the leading-order bias in R^2 but is not itself unbiased for ϕ^2.
- `phi2_umvue`: Muirhead's (1985) unique minimum variance unbiased estimator (his θ_U, their Eq. 4); equivalent to Stuart, Ord, and Arnold's (1999) equation 28.97. Requires $N \geq p + 6$ (the gate on all three Muirhead estimators); for smaller N the value is `NA`.
- `phi2_umvue_l`: Muirhead's (1985) linearly-improved unique minimum variance unbiased estimator (his θ_L); equivalent to Stuart et al.\ (1999) equation 28.98. Dominates `phi2_umvue` in mean squared error.
- `phi2_umvue_nl`: Muirhead's (1985) nonlinearly-improved estimator (his θ_{NL}). Dominates the linear improvement in MSE but requires $p \geq 5$; for smaller p the value is `NA`.

The nonlinear estimator dominates the linear one in risk when $p \geq 5$, though Muirhead notes the two perform very similarly in practice; for smaller p the linear estimator is preferred over the plug-in and adjusted-R^2 forms. All three Muirhead estimators are reported truncated at zero, so the returned value is on the same scale as ϕ^2 (the truncation introduces a negligible bias only when the population ϕ^2 is near zero).

As N grows with p fixed, the five estimators converge to a common value (the population ϕ^2); the difference between them is the small-sample bias machinery in operation. The `@examples` block illustrates that convergence.

Value

A `data.frame` with columns `term` and `value` and one row per estimator: `phi2_hat`, `phi2_adj_hat`, `phi2_umvue`, `phi2_umvue_l`, and `phi2_umvue_nl`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*, 524–555. doi:10.1080/00273170802490632

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on R^2 as a model comparison effect size.)

Muirhead, R. J. (1985). Estimating a particular function of the multiple correlation coefficient. *Journal of the American Statistical Association, 80*, 923–925.

Stuart, A., Ord, J. K., & Arnold, S. (1999). *Kendall's advanced theory of statistics, volume 2A: Classical inference and the linear model* (6th ed.). Arnold.

See Also

`ci_R2`, `ss_aipe_R2`, `convert_R2_f`

Examples

```
# 1. Fixed R^2 = 0.5 and p = 2, growing N: the five estimators agree
#    to within a small fraction once N is moderate.
signal_to_noise_R2(R2 = .5, N = 50,   p = 2)
signal_to_noise_R2(R2 = .5, N = 100,  p = 2)
signal_to_noise_R2(R2 = .5, N = 500,  p = 2)

# 2. With p = 5 the nonlinear estimator is available; it differs
#    most from the plug-in at small N.
signal_to_noise_R2(R2 = .5, N = 50,  p = 5)
signal_to_noise_R2(R2 = .5, N = 500, p = 5)
```

`simple_effects_AB` *Simple Effect F Tests for a Two-Factor Between-Subjects Design*

Description

Given a fitted `aov` or `lm` object for a two-factor between-subjects design, conventionally written $Y \sim A \times B$, where A and B are crossed fixed factors, computes the family of *simple main effects*: the effect of A at each level of B and/or the effect of B at each level of A. Each row carries the simple effect F test, its (optionally adjusted) p-value, and the partial η^2 with a noncentrality-based confidence interval. The error term can be either the full-model pooled MS_W (the textbook default) or a Welch–Satterthwaite test refitted within each conditioning level (robust to within-level heteroscedasticity).

Usage

```
simple_effects_AB(
  object,
  which = "both",
  error_term = "pooled",
  adjust = "none",
  conf_level = 0.95
)
```

Arguments

`object` A fitted `aov` or `lm` object whose right-hand side has *exactly two* crossed factors (e.g.\ `iq_gain ~ treatment * grade`). The interaction term is strongly recommended so that the pooled error is the pure within-cell MS_W; the function still runs without it but issues a warning (the additive-model residual includes interaction variance and inflates the error term used for the pooled simple effect F).

`which` Which family of simple effects to report:

- `"both"` **(default)** The b tests of A at each level of B followed by the a tests of B at each level of A ($a + b$ rows).
- `"A_at_B"` Only the b tests of the first factor at each level of the second.
- `"B_at_A"` Only the a tests of the second factor at each level of the first.

The first factor on the right-hand side of the model formula is treated as A; the second as B.

`error_term` Error-term strategy for the simple effect F:

- `"pooled"` **(default)** Use the full factorial model's MS_W and its residual *df*. This is Maxwell, Delaney, and Kelley's preferred default and gives the simple effect F maximum denominator *df*. Validity rests on homogeneity of variance across all $a \times b$ cells.

`"welch"` Refit a Welch–Satterthwaite one-way test on only the data at the conditioning level (using `oneway.test` with `var.equal = FALSE`). Robust to heteroscedasticity within the conditioning level at the cost of fewer (and fractional) denominator *df*.

`adjust` Multiple-comparison adjustment applied to the p-values of the entire family of simple effects returned ($a + b$ for `which = "both"`). One of `"none"` (default), `"bonferroni"`, or any sequential method supported by `p.adjust`: `"holm"`, `"hochberg"`, `"BH"`, `"BY"`.

`conf_level` Confidence level for each row's partial η^2 interval. Default `0.95`.

Details

What a simple effect is. The simple main effect of A at level $B = b_j$ tests whether the a cell means at that single level of B differ. It is the one-way analysis of variance of Y on A restricted to observations with $B = b_j$. The counterpart, the simple effect of B at $A = a_i$, is defined symmetrically.

Test statistic. For the simple effect of A at $B = b_j$, let $SS_{A \mid b_j} = \sum_i n_{ij} (\bar{Y}_{ij\cdot} - \bar{Y}_{\cdot j\cdot})^2$ be the between-A sum of squares computed at that level, and let $MS_{A \mid b_j} = SS_{A \mid b_j}/(a-1)$.

- *Pooled* MS_W: $F = MS_{A \mid b_j}/MS_W$ with degrees of freedom $(a-1,\ N-ab)$, where MS_W and its *df* come from the fitted full factorial model.
- *Welch*: F and its (fractional) denominator *df* come from `oneway.test(y ~ A, subset = (B == b_j), var.equal = FALSE)`.

Test statistics for B at a_i are computed by interchanging the two factors.

Choosing an error term. The pooled denominator borrows strength from all N observations and is the textbook default in Maxwell, Delaney, and Kelley's treatment. Its validity rests on homogeneity of variance across *all* $a \times b$ cells, not merely within the conditioning level. When that assumption is doubtful , for example, if Levene's or the Brown–Forsythe test flags heteroscedasticity, or if cell variances visibly differ, the Welch option provides a level-conditional test that does not require homogeneity across cells. The trade-off is denominator *df*: pooled carries the full $N - ab$ residual *df*, whereas Welch carries the Welch–Satterthwaite *df* based on the a cell variances at that level only.

Partial η^2 and its CI. The point estimate is

$$\hat{\eta}_p^2 = \frac{df_{\text{effect}}\, F}{df_{\text{effect}}\, F + df_{\text{error}}},$$

computed from the *F* and the *df* actually used in the test (so it reflects whichever error term was chosen). The confidence interval is built by Steiger's (2004) transformation principle: a CI for the noncentrality parameter λ of the *F* distribution is obtained via `ci_nc_F` and then mapped through $\eta_p^2 = \lambda/(\lambda + N_{\text{ref}})$, with N_{ref} taken to be the total study *N* for the pooled error term (treating the simple effect as a contrast within the full factorial design) and the level-conditional sample size $n_{|b_j}$ for the Welch error term (since the Welch test uses only those observations). When the lower limit on λ is not identified (i.e., the observed *F* is below its one-sided critical value), the lower limit on η_p^2 is set to 0; when the upper limit is at infinity, the upper limit on η_p^2 is set to 1.

Multiplicity across the family. With `which = "both"` the family is the $a+b$ simple effects returned in a single call; the adjustment is applied to that entire family. If only one direction is wanted, call

the function twice with `which = "A_at_B"` and `which = "B_at_A"` so each family is adjusted on its own. The `"bonferroni"` adjustment is $p_{\text{adj}} = \min(1, m\,p)$ for m rows; the sequential methods (`"holm"`, `"hochberg"`, `"BH"`, `"BY"`) are computed via `p.adjust`.

When to perform simple effects. It is no longer required that a significant omnibus interaction precede simple effect testing (Maxwell, Delaney, & Kelley, 2027). Simple effects are informative whenever the substantive question is conditional on a level of the other factor, and the multiplicity adjustment controls the family-wise error rate independently of any interaction screen.

Scope. Only fixed-effects between-subjects designs with *exactly two* crossed factors are supported. Within-subjects or mixed designs (`aovlist` fits) and three- or higher-way designs are out of scope for this function. Per-cell sample sizes may be unequal.

Value

A `data.frame` with one row per simple effect test and columns `effect`, `focal_factor`, `conditioning_factor`, `conditioning_level`, `F_value`, `df_effect`, `df_error`, `p_value`, `p_adjusted`, `partial_eta_squared`, `lower_limit`, `upper_limit`, `n_at_level`. Attributes `error_term`, `adjust`, `conf_level`, `factor_A`, and `factor_B` record the call options.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Steiger, J. H. (2004). Beyond the *F* test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

Welch, B. L. (1951). On the comparison of several mean values: An alternative approach. *Biometrika, 38*, 330–336.

See Also

`contrast_test` for within-level pairwise or custom contrasts, `eta_squared_partial` and `ci_eta_squared_partial` for the omnibus effect size counterparts, `ci_nc_F` for the noncentrality machinery, `ss_power_factorial_anova` for power calculations on the omnibus factorial effects.

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `correlations_test()`, `equivalence_r()`, `equivalence_smd()`, `factorial_anova()`, `manova_split_plot()`, `mauchly_test()`, `mixed_anova()`, `obrien_test()`, `pairwise_within()`, `randomization_test()`, `randomization_test_paired()`, `regions_of_significance()`, `summary_t_test()`, `welch_t()`

Examples

```
# 2 x 3 factorial: expectancy treatment (A) x grade (B) on the
# pygmalion data. Grades 4 through 6 are omitted so the family of
# simple effects stays short enough to read at a glance.
pyg <- pygmalion[pygmalion$grade <= 3, ]
pyg$grade <- factor(pyg$grade)
fit <- aov(iq_gain ~ treatment * grade, data = pyg)

# Default: pooled MS_W, both families, no adjustment. The expectancy
# effect on IQ gain is concentrated in grades 1 and 2; at grade 3 the
# F is 0.004, so the lower limit on partial eta squared is clamped to
# 0 and the function notes the clamp in a warning.
simple_effects_AB(fit)

# Only the simple effects of grade within each treatment level, with
# a Holm adjustment across that family of two tests.
simple_effects_AB(fit, which = "B_at_A", adjust = "holm")

# Welch error term: refits a Welch one-way at each conditioning
# level. The Welch denominator df fall well below the pooled 157, so
# more of the lower limits are clamped to 0.
simple_effects_AB(fit, error_term = "welch")

# Bonferroni across the full a + b = 5-test family.
simple_effects_AB(fit, adjust = "bonferroni")

# Simulated 2 x 2 design with a known interaction pattern.
set.seed(113)
d <- simulate_ancova_factorial_data(
  a = 2, b = 2,
  mu_y    = c(50, 60, 55, 50),   # crossover at B = 2
  mu_x    = matrix(10, nrow = 4, ncol = 1),
  sigma_y = 8, sigma_x = 3, rho_y_x = 0,
  n       = 30
)
fit_sim <- aov(y ~ A * B, data = d)
simple_effects_AB(fit_sim, conf_level = 0.95)
```

`simple_structure` *Quantify Simple Structure in a Factor Loading Matrix*

Description

Summarizes how closely a rotated loading matrix approaches Thurstone's simple structure, in which each item loads on as few factors as possible so that the factors are interpretable. Three complementary quantities are reported: the mean item complexity (the average number of factors an item effectively loads on, one for a perfectly simple item), the hyperplane proportion (the share of loadings near zero, which Thurstone sought to maximize), and the counts of pure versus complex items at a salience cutoff. Together they turn a visual impression of a loading matrix into numbers.

Usage

```
simple_structure(Lambda, salient = 0.3, hyperplane = 0.1)
```

Arguments

`Lambda`	A numeric matrix of factor loadings, items in rows and factors in columns (for example `unclass(psych::fa(...)$loadings)` or a lavaan standardized loading matrix). Row names, if present, label the items.
`salient`	Absolute loading at or above which an item is counted as loading saliently on a factor. Defaults to 0.30 (about ten percent of an item's variance), a common floor for a meaningful loading.
`hyperplane`	Absolute loading below which a loading is treated as lying in the hyperplane (effectively zero). Defaults to 0.10.

Details

Item complexity is Hofmann's complexity index, proposed in Hofmann (1977) and given as Equation 1 of Hofmann (1978), $c_i = (\sum_j \lambda_{ij}^2)^2 / \sum_j \lambda_{ij}^4$, which equals one when an item loads on a single factor and rises toward the number of factors as the loadings spread out; it is the same complexity that `psych::fa` reports. The `"complexity"` attribute holds these per-item values, and the `mean_complexity` row is their arithmetic average, what Hofmann (1978) calls the total matrix complexity. These are complexities, not Kaiser's (1974) simplicity index; Hofmann (1978) shows that either can be derived from the other at the item level, with the simplicity of item i in an m-factor solution given by his Equation 3, $s_i = [1/(m-1)][(m/c_i) - 1]$. An item is *pure* when exactly one of its loadings is salient and *complex* when more than one is. The hyperplane proportion is the fraction of all loadings whose absolute value is below `hyperplane`; a clean simple structure is mostly such near-zero loadings.

Value

A `data.frame` (class `dmar_tbl`) with one row per summary quantity (`term`, `value`): the number of items and factors, the mean and median item complexity, the hyperplane proportion, and the counts and proportion of pure items. The per-item complexities are attached as the `"complexity"` attribute (a named numeric vector), and the salience and hyperplane cutoffs as the `"salient"` and `"hyperplane"` attributes.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Hofmann, R. J. (1977). Indices descriptive of factor complexity. *The Journal of General Psychology, 96*, 58–66.

Hofmann, R. J. (1978). Complexity and simplicity as objective indices descriptive of factor solutions. *Multivariate Behavioral Research, 13*(2), 247–250.

Kaiser, H. F. (1974). An index of factorial simplicity. *Psychometrika, 39*(1), 31–36.

Thurstone, L. L. (1947). *Multiple-factor analysis*. University of Chicago Press.

See Also

average_variance_extracted and htmt for the convergent and discriminant sides of an exploratory solution.

Other multivariate and latent variable methods: average_variance_extracted(), bifactor_indices(), cfa_1(), cfa_2(), cfa_k(), ci_eigenvalue(), common_method_marker(), common_method_single_factor(), dmacs(), ecvi(), htmt(), irt_grm(), irt_information(), measurement_alignment(), measurement_invariance(), procrustes_phi()

Examples

```
# A nearly simple two-factor structure: six items, three per factor.
Lambda <- rbind(
  i1 = c(0.80, 0.05), i2 = c(0.75, 0.10), i3 = c(0.70, -0.05),
  i4 = c(0.08, 0.78), i5 = c(-0.04, 0.72), i6 = c(0.30, 0.60))
simple_structure(Lambda)
attr(simple_structure(Lambda), "complexity")
```

---

simulate_ancova_data *Simulate Data From a One-Covariate ANCOVA Model*

Description

Generates random data appropriate for an analysis of covariance with one continuous outcome (Y) and one continuous covariate (X) crossed with a fixed groups. The covariate is treated as a random variable; Y and X are jointly multivariate normal within each group. Both the randomized-design case (population covariate mean common across groups) and the non-randomized / preexisting-groups case (population covariate means differ across groups; Y-on-X correlation may also differ) are supported. Per-group sample sizes may be equal or unequal.

Usage

```
simulate_ancova_data(
  mu_y,
  mu_x,
  sigma_y,
  sigma_x,
  rho,
  a,
  n,
  randomized = TRUE
)
```

Arguments

mu_y	A numeric vector of length a giving the population mean of Y in each group.
mu_x	When randomized = TRUE, a single number giving the common population mean of X across all groups. When randomized = FALSE, a numeric vector of length a giving the population covariate mean in each group.
sigma_y	The population standard deviation of Y (assumed common across groups).
sigma_x	The population standard deviation of X (assumed common across groups).
rho	The population correlation between Y and X. When randomized = TRUE, must be a single number (see Details). When randomized = FALSE, may be a single number (recycled to all a groups) or a numeric vector of length a giving a distinct correlation in each group.
a	The number of fixed levels of the grouping factor (i.e., the number of conditions in a fixed-effects ANCOVA design). Use this argument when groups are the design levels of interest. (For sample-selected groups, e.g., classrooms or schools randomly drawn from a population, the convention in this package is to use J instead.)
n	A single number (equal sample size per group) or a numeric vector of length a giving the sample size in each group.
randomized	Logical. TRUE (the default) for a randomized design (random assignment of subjects to groups, so that the population covariate mean is the same in every group). FALSE for a non-randomized / preexisting-groups design.

Details

Each group's (Y, X) pairs are drawn from a bivariate normal distribution with mean $(\mu_{Y,j}, \mu_{X,j})$ and covariance

$$\Sigma_j = \begin{pmatrix} \sigma_Y^2 & \rho_j\, \sigma_Y\, \sigma_X \\ \rho_j\, \sigma_Y\, \sigma_X & \sigma_X^2 \end{pmatrix}.$$

Why rho must be a single number when randomized = TRUE. Random assignment forms each group as an exchangeable random sample from the same population. The bivariate distribution of (Y, X) is therefore the same in every group, including the correlation. Allowing rho to differ across groups would silently break that interpretation and produce data that no randomized design could plausibly have generated. The function therefore stops with an error in that case; use randomized = FALSE if you genuinely want group-specific correlations.

Convention on group labels. The argument a is used here (and throughout DMAR's experimental-design functions) for the number of *fixed* levels of a designed factor, the levels you intend to compare. The letter J is reserved for the number of *sample- selected* groups, e.g., when classrooms or schools are randomly sampled from a population (a random-effects context).

Value

A long-format data.frame with one row per simulated subject and three columns:

group A factor with a levels ("1", ..., as.character(a)) identifying each subject's group.

y Numeric simulated outcome.

x Numeric simulated covariate.

This format is directly usable with aov(), lm(), and other model-fitting functions.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

mvrnorm, ci_c_ancova, ss_aipe_c_ancova

Other data simulators: simulate_ancova_factorial_data(), simulate_anova_data(), simulate_longitudinal_gomp
simulate_longitudinal_logistic(), simulate_longitudinal_negative_exponential(), simulate_longitudinal_
simulate_longitudinal_richards(), simulate_regression_data()

Examples

```
# 1. Randomized design, two groups, equal n.
set.seed(113)
simple <- simulate_ancova_data(
  mu_y       = c(3, 5),
  mu_x       = 10,
  sigma_y    = 1,
  sigma_x    = 2,
  rho        = 0.8,
  a          = 2,
  n          = 20
)
head(simple)

# 2. Four preexisting groups with different correlations and unequal n.
#    The first two groups share rho = 0.30; the second two share a larger
#    rho = 0.60. The four groups are not used in the data-generation
#    machinery beyond their per-group means and correlations -- they are
#    just four distinct populations being sampled. In a downstream
#    analysis these four groups could be cross-classified as a 2 x 2
#    factorial design (e.g., the first factor distinguishing groups 1-2
#    from groups 3-4, and the second factor distinguishing groups 1, 3
#    from groups 2, 4) and analyzed via factorial ANCOVA.
set.seed(113)
preexisting <- simulate_ancova_data(
  mu_y       = c(50, 55, 60, 65),
  mu_x       = c(10, 12, 11, 13),
  sigma_y    = 8,
  sigma_x    = 3,
  rho        = c(0.30, 0.30, 0.60, 0.60),
  a          = 4,
  n          = c(40, 35, 45, 30),
  randomized = FALSE
)
aggregate(cbind(y, x) ~ group, data = preexisting,
          FUN = function(z) round(c(mean = mean(z), sd = sd(z)), 2))
```

`simulate_ancova_factorial_data` *Simulate Data From a Factorial ANCOVA Design (up to Four Factors, Any Number of Covariates)*

Description

Generates random data appropriate for an analysis of covariance with up to four crossed fixed factors (A, B, C, D) and one or more continuous covariates. Within every cell the outcome Y and the covariates $X_1, \ldots, X_q$ are jointly multivariate normal with a common (homogeneous) within-cell covariance structure, the standard assumption underlying classical ANCOVA. Per-cell sample sizes may be equal or unequal.

Usage

```
simulate_ancova_factorial_data(
  a,
  b = 1,
  c = 1,
  d = 1,
  n_covariates = 1,
  mu_y,
  mu_x,
  sigma_y,
  sigma_x,
  rho_y_x,
  rho_x_x = NULL,
  n,
  randomized = TRUE
)
```

Arguments

`a`	Number of levels of the first factor (must be at least `2`).
`b`	Number of levels of the second factor (default `1`; use `1` if absent).
`c`	Number of levels of the third factor (default `1`; use `1` if absent).
`d`	Number of levels of the fourth factor (default `1`; use `1` if absent). So `a = 3, b = 2, c = 1, d = 1` specifies a 3×2 design.
`n_covariates`	Integer ≥ 1: the number of continuous covariates. Default `1`.
`mu_y`	Numeric vector of length $a \cdot b \cdot c \cdot d$ giving the population cell means of Y, in the same row order as `expand.grid(A, B, C, D)` (which varies factor `A` fastest, then `B`, then `C`, then `D`).

mu_x	Numeric matrix of dimension $(a \cdot b \cdot c \cdot d) \times q$ giving the population cell means of each covariate (rows = cells in the same order as mu_y; columns = covariates). When randomized = TRUE every column must be constant across cells.
sigma_y	Within-cell standard deviation of Y (a single positive number, common across cells).
sigma_x	Within-cell standard deviations of the covariates. Either a single number (recycled to all n_covariates) or a numeric vector of length n_covariates.
rho_y_x	Within-cell correlations between Y and each covariate. Either a single number (recycled to all n_covariates) or a numeric vector of length n_covariates.
rho_x_x	Within-cell correlation matrix among the covariates, $q \times q$. Default NULL, interpreted as the n_covariates-dimensional identity.
n	A single number (equal sample size per cell) or a numeric vector of length $a \cdot b \cdot c \cdot d$ giving per-cell sample sizes (in the same row order as mu_y).
randomized	Logical. TRUE (the default) for a randomized design, each cell is sampled from the same population covariate distribution, so mu_x must be constant across cells. FALSE for non-randomized / preexisting-groups designs in which the cells' population covariate means may differ.

Details

This is the factorial generalization of simulate_ancova_data (which is the special case b = c = d = 1, n_covariates = 1). All cells share the same within-cell covariance structure (homogeneity of regression, the classical ANCOVA assumption); the difference between randomized and non-randomized designs lies entirely in whether the cell covariate means are constrained to be equal.

Why mu_x must be constant across cells when randomized = TRUE. Random assignment forms each cell as an exchangeable random sample from the same joint distribution of covariates and outcome. Cell-specific covariate means would silently break that interpretation. The function checks the constraint and stops with an informative error if it is violated. (The same logic that powers simulate_ancova_data.)

Cell ordering. The function uses expand.grid's convention, factor A varies fastest, then B, then C, then D. So for a 2×3 design, the six cells of mu_y are $(A_1B_1), (A_2B_1), (A_1B_2), (A_2B_2), (A_1B_3), (A_2B_3)$. If you build the cell specification by passing the factor levels to expand.grid in the same order, the row indexing automatically matches.

Value

A long-format data.frame with one row per simulated subject and the following columns:

A, B, C, D Factor columns for each present design factor (omitted when the factor is absent, i.e., its corresponding a/b/c/d argument is 1).

x1, x2, ..., x<q> Numeric simulated covariates.

y Numeric simulated outcome.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

simulate_ancova_data (one factor, one covariate special case), simulate_anova_data, simulate_regression_data

Other data simulators: simulate_ancova_data(), simulate_anova_data(), simulate_longitudinal_gompertz(), simulate_longitudinal_logistic(), simulate_longitudinal_negative_exponential(), simulate_longitudinal_richards(), simulate_regression_data()

Examples

```
# 1. 2 x 2 randomized design, single covariate.
set.seed(113)
design_2x2 <- expand.grid(A = factor(1:2), B = factor(1:2))
design_2x2$mu_y <- c(50, 60, 55, 65)   # cell means in expand.grid order

d1 <- simulate_ancova_factorial_data(
  a            = 2, b = 2,
  mu_y         = design_2x2$mu_y,
  mu_x         = matrix(10, nrow = 4, ncol = 1),  # constant covariate mean
  sigma_y      = 8,
  sigma_x      = 3,
  rho_y_x      = 0.40,
  n            = 30
)
aggregate(y ~ A + B, data = d1, FUN = mean)

# 2. 3 x 2 nonrandomized design with two covariates and unequal n.
set.seed(113)
a <- 3; b <- 2; q <- 2
n_cells <- a * b
d2 <- simulate_ancova_factorial_data(
  a            = a, b = b,
  n_covariates = q,
  mu_y         = c(50, 55, 60,  52, 58, 64),
  mu_x         = matrix(c(10, 11, 12,  9, 10, 11,
                          5,  6,  7,  4,  5,  6),
                        nrow = n_cells, ncol = q),
  sigma_y      = 8,
  sigma_x      = c(3, 2),
  rho_y_x      = c(0.40, 0.25),
  rho_x_x      = matrix(c(1,   0.3,
                          0.3, 1),
                        nrow = 2),
  n            = c(30, 25, 20,  35, 30, 25),
  randomized   = FALSE
)
head(d2)

# 3. 2 x 2 x 2 randomized design with one covariate.
set.seed(113)
d3 <- simulate_ancova_factorial_data(
  a       = 2, b = 2, c = 2,
  mu_y    = c(50, 55, 52, 57,  53, 58, 55, 60),  # 8 cells in A-fastest order
```

```
  mu_x    = matrix(10, nrow = 8, ncol = 1),
  sigma_y = 8,
  sigma_x = 3,
  rho_y_x = 0.40,
  n       = 25
)
table(d3$A, d3$B, d3$C)  # 25 per cell, 200 total
```

`simulate_anova_data` *Simulate Data From a One-Way Fixed-Effects ANOVA Model*

Description

Generates random data appropriate for a one-way fixed-effects analysis of variance. Each group's observations are drawn from a normal distribution with that group's population mean and a common (or per-group) standard deviation. Per-group sample sizes may be equal or unequal.

Usage

```
simulate_anova_data(mu, sigma, a, n, seed = NULL)
```

Arguments

`mu`	A numeric vector of length `a` giving the population mean in each group.
`sigma`	Within-group population standard deviation. Either a single number (homoscedastic; common across groups) or a numeric vector of length `a` (heteroscedastic; one SD per group).
`a`	The number of fixed levels of the grouping factor (per the convention used throughout DMAR for fixed-factor designs).
`n`	A single number (equal sample size per group) or a numeric vector of length `a` giving the sample size in each group.
`seed`	Optional integer random seed for reproducibility (default `NULL`; supply an integer such as `113` for reproducible output).

Details

The fixed-effects ANOVA model assumes group-specific means and a common within-group variance. Setting `sigma` to a vector relaxes the homoscedasticity assumption; in that case the simulated data violate the standard ANOVA assumption (a useful feature for studying robustness or the performance of Welch-style alternatives).

Value

A long-format `data.frame` with one row per simulated subject and two columns:

`group` A factor with `a` levels.

`y` Numeric simulated outcome.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

simulate_ancova_data, simulate_regression_data, contrast_test, ss_power_contrast

Other data simulators: simulate_ancova_data(), simulate_ancova_factorial_data(), simulate_longitudinal_gom
simulate_longitudinal_logistic(), simulate_longitudinal_negative_exponential(), simulate_longitudinal_
simulate_longitudinal_richards(), simulate_regression_data()

Examples

```
# Three-group ANOVA, equal n per group.
set.seed(113)
d <- simulate_anova_data(mu = c(50, 55, 60), sigma = 8, a = 3, n = 30)
aggregate(y ~ group, data = d, FUN = mean)

# Same design with unequal n per group.
simulate_anova_data(mu = c(50, 55, 60), sigma = 8, a = 3,
                    n = c(40, 30, 20), seed = 113)

# Heteroscedastic case: each group has its own SD.
simulate_anova_data(mu = c(50, 55, 60), sigma = c(5, 8, 12),
                    a = 3, n = 30, seed = 113)
```

---

simulate_longitudinal_gompertz

Simulate Data From a Gompertz Change (Growth) Model

Description

Generates longitudinal data from a random-coefficients Gompertz change model: each unit (a person, an animal, a tree) follows an S-shaped curve that, unlike the logistic, is not symmetric about its point of inflection, the parameters vary randomly across units, and each measurement adds level-one error. The deterministic part of the four parameter Gompertz curve is

$$\mu(t) = \alpha \exp\left(-\exp(-\gamma(t-\beta))\right) + \zeta,$$

the parameterization of Kelley (2005, 2008), which generalizes the three parameter Gompertz of the literature (Winsor, 1932; Ratkowsky, 1983) by adding ζ so the lower asymptote is itself a modeled quantity rather than fixed at zero.

Usage

```
simulate_longitudinal_gompertz(
  n,
  target_times = NULL,
  fixed_parameters,
  time_range = NULL,
  occasions = NULL,
  time_distribution = "uniform",
  random_variances = 0,
  random_correlation = NULL,
  error_variance = NULL,
  reliability = NULL,
  error_structure = c("independent", "ar1", "compound_symmetry", "toeplitz"),
  error_correlation = NULL,
  timing_sd = 0
)
```

Arguments

n	A single positive integer, the number of units (persons, animals, trees, classrooms) whose trajectories are drawn, or a vector giving the number of units for each population, one entry per parameter vector in `fixed_parameters`.
target_times	Numeric vector of the nominal measurement times, one shared schedule for every unit. Give either this or `time_range`.
fixed_parameters	The population parameters `c(alpha = , beta = , gamma = , zeta = )` (an unnamed length-4 vector is taken in that order), or a list of such vectors, one per population (distinct data generating parameter vectors): `alpha` The total change: the curve travels `alpha` units from the lower asymptote `zeta` to the upper asymptote $\alpha + \zeta$ (for $\gamma > 0$). `beta` The point of inflection on the time axis. The Gompertz inflection is early and asymmetric: it occurs where the curve passes through $\alpha/e + \zeta$, about 36.8% of the total change, so growth accelerates briefly and decelerates over a long approach to the ceiling. `gamma` The curvature: how sharply the curve rises through its inflection. Negative values flip the curve to decreasing. `zeta` The lower asymptote (for $\gamma > 0$). The intercept is $\phi = \alpha \exp(-\exp(\gamma\beta)) + \zeta$.
time_range	Alternative to `target_times`: `c(lower,upper)` bounds from which each unit draws its own measurement times, so no two units share a schedule (e.g., age in weeks at testing rather than a fixed grade). Requires `occasions`; with `time_range`, the level-one error must be a single `error_variance` with the default independent structure, and `timing_sd` does not apply.
occasions	With `time_range`: a single positive integer (every unit measured the same number of times) or `c(min,max)`, from which each unit's number of measurement times is drawn uniformly.

`time_distribution`
: Distribution of the unit-specific times over `time_range`; currently `"uniform"`.

`random_variances`
: Between-unit variances of `(alpha, beta, gamma, zeta)`, a single number recycled to all four or a length-4 vector. Default `0`. A named vector over any subset of the parameter names (e.g., `c(beta = 1.2)`) varies only those named and leaves the rest fixed.

`random_correlation`
: Optional 4-by-4 correlation matrix among the random parameters; default uncorrelated.

`error_variance`, `reliability`, `error_structure`, `error_correlation`, `timing_sd`
: The level-one error and assessment-time machinery, with the same meaning as in `simulate_longitudinal_polynomial`: specify exactly one of `error_variance` or `reliability` (solved here through the first-order delta method true-score variance; because the curve is nonlinear in its parameters, that approximation, and the `reliability_by_occasion` attribute with it, can drift from the realized variance ratio when the random variances are large relative to the mean curve); `error_structure` and `error_correlation` set the across-occasion error correlation; `timing_sd` jitters the actual assessment times.

Details

The choice between the Gompertz and the logistic is substantive, not cosmetic: both are S-shaped, but the logistic spends equal time approaching floor and ceiling while the Gompertz commits to an early inflection (36.8% of total change) followed by a long deceleration. Processes with rapid early gains and slow consolidation, common in learning and development, are natural Gompertz candidates. The Gompertz is the $\delta \to 0$ limit of the Richards curve (`simulate_longitudinal_richards`), which frees the inflection entirely (Kelley, 2005, 2008).

Value

A long-format `data.frame` with columns `id`, `population`, `occasion`, `target_time`, `time`, `true_score`, and `y`, directly usable with `plot_trajectories` and nonlinear mixed-model fitters such as `nlme::nlme()`. Attributes carry the `model`, `fixed_parameters`, `random_covariance`, `error_variance`, `error_covariance`, `reliability_by_occasion`, and `schedule` (`"shared"` or `"unit_specific"`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2005). *Estimating nonlinear change models in heterogeneous populations when class membership is unknown: Defining and developing the latent classification differential change model* (Doctoral dissertation). University of Notre Dame.

Kelley, K. (2008). Nonlinear change models in populations with unobserved heterogeneity. *Methodology, 4*(3), 97–112.

Ratkowsky, D. A. (1983). *Nonlinear regression modeling: A unified practical approach*. Marcel Dekker.

Winsor, C. P. (1932). The Gompertz curve as a growth curve. *Proceedings of the National Academy of Sciences, 18*(1), 1–8.

See Also

simulate_longitudinal_logistic for the symmetric sibling; simulate_longitudinal_richards for the family that subsumes both; simulate_longitudinal_negative_exponential; simulate_longitudinal_polynom plot_trajectories.

Other data simulators: simulate_ancova_data(), simulate_ancova_factorial_data(), simulate_anova_data(), simulate_longitudinal_logistic(), simulate_longitudinal_negative_exponential(), simulate_longitudinal_ simulate_longitudinal_richards(), simulate_regression_data()

Examples

```
# The six-curve illustration from Kelley (2005): alpha = 0.75 and
# zeta = 0.25 throughout, so every curve crosses its inflection at
# the same height, 0.75 / exp(1) + 0.25, about 0.526. The three
# rising curves share the inflection time beta = 2 and differ only
# in curvature; the three falling curves share beta = 3. Each curve
# is its own population of size one, so a single call draws the whole panel.
panel <- simulate_longitudinal_gompertz(
  n = 1, target_times = seq(0, 6, by = 0.1),
  fixed_parameters = list(
    c(alpha = 0.75, beta = 2, gamma =  1.75, zeta = 0.25),
    c(alpha = 0.75, beta = 2, gamma =  1.00, zeta = 0.25),
    c(alpha = 0.75, beta = 2, gamma =  0.45, zeta = 0.25),
    c(alpha = 0.75, beta = 3, gamma = -0.35, zeta = 0.25),
    c(alpha = 0.75, beta = 3, gamma = -0.60, zeta = 0.25),
    c(alpha = 0.75, beta = 3, gamma = -2.00, zeta = 0.25)),
  error_variance = 0
)
plot_trajectories(panel, id = "id", time = "time",
                  outcome = "true_score", group = "population")

# Individual differences in a single parameter: only the inflection
# time varies (a named entry leaves every other variance at zero),
# so every trajectory shares the floor and the ceiling but reaches
# its fastest growth at its own moment.
set.seed(113)
d_beta <- simulate_longitudinal_gompertz(
  n = 25, target_times = seq(0, 8, by = 0.5),
  fixed_parameters = c(alpha = 75, beta = 3, gamma = 0.55, zeta = 10),
  random_variances = c(beta = 0.8), error_variance = 0
)
plot_trajectories(d_beta, id = "id", time = "time",
                  outcome = "true_score")

# Individual differences in every parameter at once, plus level-one
# error: the realistic sampling model.
```

```
set.seed(113)
d <- simulate_longitudinal_gompertz(
  n = 30, target_times = 0:12,
  fixed_parameters = c(alpha = 75, beta = 3, gamma = 0.55, zeta = 10),
  random_variances = c(alpha = 36, beta = 0.8, gamma = 0.01, zeta = 9),
  error_variance = 16
)
plot_trajectories(d, id = "id", time = "time", outcome = "y")
```

`simulate_longitudinal_logistic` *Simulate Data From a Logistic Change (Growth) Model*

Description

Generates longitudinal data from a random-coefficients logistic change model: each unit (a person, an animal, a tree) follows an S-shaped (sigmoidal) curve with a lower and an upper asymptote and a symmetric point of inflection, the parameters vary randomly across units, and each measurement adds level-one error. The deterministic part of the four parameter logistic curve is

$$\mu(t) = \frac{\alpha}{1 + \exp(-\gamma(t - \beta))} + \zeta,$$

the parameterization of Kelley (2005, 2008), which generalizes the three parameter logistic of the literature (Ratkowsky, 1983) by adding ζ so the lower asymptote is itself a modeled quantity rather than fixed at zero.

Usage

```
simulate_longitudinal_logistic(
  n,
  target_times = NULL,
  fixed_parameters,
  time_range = NULL,
  occasions = NULL,
  time_distribution = "uniform",
  random_variances = 0,
  random_correlation = NULL,
  error_variance = NULL,
  reliability = NULL,
  error_structure = c("independent", "ar1", "compound_symmetry", "toeplitz"),
  error_correlation = NULL,
  timing_sd = 0
)
```

Arguments

`n` A single positive integer, the number of units (persons, animals, trees, classrooms) whose trajectories are drawn, or a vector giving the number of units for each population, one entry per parameter vector in `fixed_parameters`.

`target_times` Numeric vector of the nominal measurement times, one shared schedule for every unit. Give either this or `time_range`.

`fixed_parameters`
The population parameters `c(alpha = , beta = , gamma = , zeta = )` (an unnamed length-4 vector is taken in that order), or a list of such vectors, one per population (distinct data generating parameter vectors):

- `alpha` The total change: the curve travels `alpha` units from the lower asymptote `zeta` to the upper asymptote $\alpha + \zeta$ (for $\gamma > 0$).
- `beta` The point of inflection on the time axis: the moment of fastest change, at which exactly half the total change has occurred (the curve passes through $\alpha/2 + \zeta$). The inflection of the logistic is symmetric: the approach to the ceiling mirrors the departure from the floor.
- `gamma` The curvature: how sharply the curve rises through its inflection. Negative values flip the curve to decreasing.
- `zeta` The lower asymptote (for $\gamma > 0$), freeing the curve's floor from the fixed zero of the three parameter logistic. The intercept is $\phi = \alpha/(1+\exp(\gamma\beta))+\zeta$.

`time_range` Alternative to `target_times`: `c(lower,upper)` bounds from which each unit draws its own measurement times, so no two units share a schedule (e.g., age in weeks at testing rather than a fixed grade). Requires `occasions`; with `time_range`, the level-one error must be a single `error_variance` with the default independent structure, and `timing_sd` does not apply.

`occasions` With `time_range`: a single positive integer (every unit measured the same number of times) or `c(min,max)`, from which each unit's number of measurement times is drawn uniformly.

`time_distribution`
Distribution of the unit-specific times over `time_range`; currently `"uniform"`.

`random_variances`
Between-unit variances of `(alpha, beta, gamma, zeta)`, a single number recycled to all four or a length-4 vector. Default `0`. A named vector over any subset of the parameter names (e.g., `c(beta = 1.2)`) varies only those named and leaves the rest fixed.

`random_correlation`
Optional 4-by-4 correlation matrix among the random parameters; default uncorrelated.

`error_variance`, `reliability`, `error_structure`, `error_correlation`, `timing_sd`
The level-one error and assessment-time machinery, with the same meaning as in `simulate_longitudinal_polynomial`: specify exactly one of `error_variance` or `reliability` (solved here through the first-order delta method true-score variance; because the curve is nonlinear in its parameters, that approximation, and the `reliability_by_occasion` attribute with it, can drift from the realized

variance ratio when the random variances are large relative to the mean curve); `error_structure` and `error_correlation` set the across-occasion error correlation; `timing_sd` jitters the actual assessment times.

Details

Every parameter is a landmark of the change process: the floor (`zeta`), the ceiling ($\alpha + \zeta$), when change is fastest (`beta`), and how concentrated the change is around that moment (`gamma`). The logistic is the $\delta = 1$ special case of the Richards curve (`simulate_longitudinal_richards`); its inflection always sits at 50% of the total change, which is the substantive assumption a researcher accepts in choosing it (Kelley, 2005, 2008).

Value

A long-format `data.frame` with columns `id`, `population`, `occasion`, `target_time`, `time`, `true_score`, and `y`, directly usable with `plot_trajectories` and nonlinear mixed-model fitters such as `nlme::nlme()`. Attributes carry the `model`, `fixed_parameters`, `random_covariance`, `error_variance`, `error_covariance`, `reliability_by_occasion`, and `schedule` (`"shared"` or `"unit_specific"`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2005). *Estimating nonlinear change models in heterogeneous populations when class membership is unknown: Defining and developing the latent classification differential change model* (Doctoral dissertation). University of Notre Dame.

Kelley, K. (2008). Nonlinear change models in populations with unobserved heterogeneity. *Methodology, 4*(3), 97–112.

Ratkowsky, D. A. (1983). *Nonlinear regression modeling: A unified practical approach*. Marcel Dekker.

See Also

`simulate_longitudinal_gompertz` for the asymmetric sibling; `simulate_longitudinal_richards` for the family that subsumes both; `simulate_longitudinal_negative_exponential`; `simulate_longitudinal_polynom` `plot_trajectories`.

Other data simulators: `simulate_ancova_data()`, `simulate_ancova_factorial_data()`, `simulate_anova_data()`, `simulate_longitudinal_gompertz()`, `simulate_longitudinal_negative_exponential()`, `simulate_longitudinal_` `simulate_longitudinal_richards()`, `simulate_regression_data()`

Examples

```
# A six-curve panel in the style of the growth-curve illustrations
# in Kelley (2005): floor 10 and ceiling 90 throughout, so every
# curve crosses its inflection at the same height (50, half the
# total change). The rising curves share beta = 6 and differ only in
# curvature; the falling curves mirror them. Each curve is its own population
```

```
# of size one, so a single call draws the whole panel.
panel <- simulate_longitudinal_logistic(
  n = 1, target_times = seq(0, 12, by = 0.1),
  fixed_parameters = list(
    c(alpha = 80, beta = 6, gamma =  2.0, zeta = 10),
    c(alpha = 80, beta = 6, gamma =  0.9, zeta = 10),
    c(alpha = 80, beta = 6, gamma =  0.5, zeta = 10),
    c(alpha = 80, beta = 6, gamma = -0.5, zeta = 10),
    c(alpha = 80, beta = 6, gamma = -0.9, zeta = 10),
    c(alpha = 80, beta = 6, gamma = -2.0, zeta = 10)),
  error_variance = 0
)
plot_trajectories(panel, id = "id", time = "time",
                  outcome = "true_score", group = "population")

# Individual differences in a single parameter: only the inflection
# time varies (a named entry leaves the other variances at zero), so
# every learner shares the floor and the ceiling but hits fastest
# growth on a different week.
set.seed(113)
d_beta <- simulate_longitudinal_logistic(
  n = 25, target_times = seq(0, 12, by = 0.5),
  fixed_parameters = c(alpha = 80, beta = 6, gamma = 0.9, zeta = 10),
  random_variances = c(beta = 1.2), error_variance = 0
)
plot_trajectories(d_beta, id = "id", time = "time",
                  outcome = "true_score")

# Unit-specific measurement times: each child is tested at their own
# ages, drawn uniformly between 40 and 90 weeks with five to nine
# visits each, rather than on one shared schedule.
set.seed(113)
d_ages <- simulate_longitudinal_logistic(
  n = 12, time_range = c(40, 90), occasions = c(5, 9),
  fixed_parameters = c(alpha = 80, beta = 65, gamma = 0.15, zeta = 10),
  random_variances = c(beta = 16), error_variance = 4
)
plot_trajectories(d_ages, id = "id", time = "time", outcome = "y")

# Individual differences in every parameter, plus level-one error:
# skill acquisition from a floor near 10 to a ceiling near 90,
# fastest around week 6.
set.seed(113)
d <- simulate_longitudinal_logistic(
  n = 30, target_times = 0:12,
  fixed_parameters = c(alpha = 80, beta = 6, gamma = 0.9, zeta = 10),
  random_variances = c(alpha = 36, beta = 1, gamma = 0.01, zeta = 9),
  error_variance = 16
)
plot_trajectories(d, id = "id", time = "time", outcome = "y")
```

`simulate_longitudinal_negative_exponential` *Simulate Data From a Negative Exponential (Asymptotic Regression) Change Model*

Description

Generates longitudinal data from a random-coefficients negative exponential change model, also called asymptotic regression (Stevens, 1951): each unit (a person, an animal, a tree) approaches an asymptote at a rate set by a curvature parameter, the parameters vary randomly across units, and each measurement adds level-one error. The deterministic part of the curve is

$$\mu(t) = \alpha + \zeta \exp(-\gamma t),$$

the parameterization of Kelley (2005, 2008). The negative exponential is the simplest of the package's nonlinear change curves: it has one asymptote and no point of inflection, so it describes change that is fastest at the first assessment and decelerates thereafter.

Usage

```
simulate_longitudinal_negative_exponential(
  n,
  target_times = NULL,
  fixed_parameters,
  time_range = NULL,
  occasions = NULL,
  time_distribution = "uniform",
  random_variances = 0,
  random_correlation = NULL,
  error_variance = NULL,
  reliability = NULL,
  error_structure = c("independent", "ar1", "compound_symmetry", "toeplitz"),
  error_correlation = NULL,
  timing_sd = 0
)
```

Arguments

`n`	A single positive integer, the number of units (persons, animals, trees, classrooms) whose trajectories are drawn, or a vector giving the number of units for each population, one entry per parameter vector in `fixed_parameters`.
`target_times`	Numeric vector of the nominal measurement times, one shared schedule for every unit. Give either this or `time_range`.
`fixed_parameters`	The population parameters `c(alpha = , zeta = , gamma = )` (an unnamed length-3 vector is taken in that order), or a list of such vectors, one per population (distinct parameter vectors):

alpha The asymptote: the value $\mu(t)$ approaches as t grows.

zeta The negative of the total change: the curve starts at the intercept $\phi = \alpha + \zeta$ and travels $-\zeta$ units to the asymptote. Positive zeta gives asymptotic decay toward alpha from above; negative zeta gives asymptotic growth from below.

gamma The curvature ($\gamma > 0$): the rate at which the remaining distance to the asymptote closes per unit time. Larger values reach the asymptote sooner.

time_range Alternative to target_times: c(lower,upper) bounds from which each unit draws its own measurement times, so no two units share a schedule (e.g., age in weeks at testing rather than a fixed grade). Requires occasions; with time_range, the level-one error must be a single error_variance with the default independent structure, and timing_sd does not apply.

occasions With time_range: a single positive integer (every unit measured the same number of times) or c(min,max), from which each unit's number of measurement times is drawn uniformly.

time_distribution
Distribution of the unit-specific times over time_range; currently "uniform".

random_variances
Between-unit variances of (alpha, zeta, gamma), a single number recycled to all three or a length-3 vector. Default 0 (a common curve for everyone). A named vector over any subset of the parameter names (e.g., c(gamma = 0.02)) varies only those named and leaves the rest fixed.

random_correlation
Optional 3-by-3 correlation matrix among the random parameters; default uncorrelated.

error_variance, reliability, error_structure, error_correlation, timing_sd
The level-one error and assessment-time machinery, with the same meaning as in simulate_longitudinal_polynomial: specify exactly one of error_variance (scalar, per-occasion vector, or full covariance matrix) or reliability (a target average per-occasion reliability, solved by the delta method here); error_structure and error_correlation set the across-occasion error correlation; timing_sd jitters each unit's actual assessment times around the nominal targets.

Details

Every parameter answers a substantive question: where does change end (alpha), where does it start ($\phi = \alpha + \zeta$), and how fast does the gap close (gamma)? That interpretability is the argument for nonlinear change models over polynomials, whose coefficients describe no landmark of the process (Kelley, 2005, 2008); the package vignette on nonlinear growth develops the comparison.

Value

A long-format data.frame with columns id, population, occasion, target_time, time, true_score, and y, directly usable with plot_trajectories and nonlinear mixed-model fitters such as nlme::nlme(). Attributes carry the model, the fixed_parameters, the between-unit covariance random_covariance, the level-one error_variance and error_covariance, and reliability_by_occasion (from

the first-order delta method true-score variance, which can drift from the realized variance ratio when the random variances are large relative to the mean curve). The `schedule` attribute records `"shared"` or `"unit_specific"`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2005). *Estimating nonlinear change models in heterogeneous populations when class membership is unknown: Defining and developing the latent classification differential change model* (Doctoral dissertation). University of Notre Dame.

Kelley, K. (2008). Nonlinear change models in populations with unobserved heterogeneity. *Methodology, 4*(3), 97–112.

Stevens, W. L. (1951). Asymptotic regression. *Biometrics, 7*(3), 247–267.

See Also

`simulate_longitudinal_logistic`, `simulate_longitudinal_gompertz`, `simulate_longitudinal_richards` for the sigmoidal members of the family; `simulate_longitudinal_polynomial` for the polynomial counterpart; `plot_trajectories` for plotting the result.

Other data simulators: `simulate_ancova_data()`, `simulate_ancova_factorial_data()`, `simulate_anova_data()`, `simulate_longitudinal_gompertz()`, `simulate_longitudinal_logistic()`, `simulate_longitudinal_polynomial()` `simulate_longitudinal_richards()`, `simulate_regression_data()`

Examples

```
# The six-curve illustration from Kelley (2005): three growth curves
# (intercept 0, asymptote 1) that differ only in curvature, and
# three decay curves (intercept 1, asymptote 0) that mirror them.
# Each curve is its own population of size one, so a single call draws the
# whole panel.
panel <- simulate_longitudinal_negative_exponential(
  n = 1, target_times = seq(0, 10, by = 0.1),
  fixed_parameters = list(
    c(alpha = 1, zeta = -1, gamma = 0.9),
    c(alpha = 1, zeta = -1, gamma = 0.4),
    c(alpha = 1, zeta = -1, gamma = 0.2),
    c(alpha = 0, zeta =  1, gamma = 1.2),
    c(alpha = 0, zeta =  1, gamma = 0.5),
    c(alpha = 0, zeta =  1, gamma = 0.3)),
  error_variance = 0
)
plot_trajectories(panel, id = "id", time = "time",
                  outcome = "true_score", group = "population")

# Individual differences in a single parameter: only the curvature
# varies (a named entry leaves the other variances at zero), so all
# trajectories share their start and their destination but close the
```

```
# gap at their own rates.
set.seed(113)
d_gamma <- simulate_longitudinal_negative_exponential(
  n = 25, target_times = seq(0, 8, by = 0.5),
  fixed_parameters = c(alpha = 100, zeta = -80, gamma = 0.5),
  random_variances = c(gamma = 0.02), error_variance = 0
)
plot_trajectories(d_gamma, id = "id", time = "time",
                  outcome = "true_score")

# Individual differences in every parameter, plus level-one error:
# vocabulary learning that starts near 20 words (phi = alpha + zeta),
# climbs toward an asymptote near 100, and closes about 40% of the
# remaining gap per month (gamma = 0.5).
set.seed(113)
d <- simulate_longitudinal_negative_exponential(
  n = 30, target_times = 0:8,
  fixed_parameters = c(alpha = 100, zeta = -80, gamma = 0.5),
  random_variances = c(alpha = 25, zeta = 16, gamma = 0.01),
  error_variance = 9
)
head(d)
plot_trajectories(d, id = "id", time = "time", outcome = "y")
```

`simulate_longitudinal_polynomial`

Simulate Data From a Polynomial Change (Growth) Model

Description

Generates longitudinal data from a random-coefficients polynomial change model: each subject follows a degree-P polynomial in time whose coefficients vary randomly across subjects, and each measurement adds independent level-one error. The polynomial order is general (order 0 is a flat line, 1 a straight line, 2 a quadratic, and so on), one or several populations may differ in their mean trajectories, the level-one error can be set directly or pinned to a target measurement reliability, and the actual time of each assessment may jitter away from its nominal target. This is the Monte Carlo companion to `ss_power_pcm`, which plans power for the same model under the closed-form assumptions of Raudenbush and Liu (2001); the simulator can relax those assumptions (notably the assumption of fixed, error-free, equally reliable assessment times) and study what happens.

Usage

```
simulate_longitudinal_polynomial(
  n,
  target_times = NULL,
  fixed_coefficients,
  time_range = NULL,
  occasions = NULL,
```

```
  time_distribution = "uniform",
  random_variances = 0,
  random_correlation = NULL,
  error_variance = NULL,
  reliability = NULL,
  error_structure = c("independent", "ar1", "compound_symmetry", "toeplitz"),
  error_correlation = NULL,
  timing_sd = 0
)
```

Arguments

`n`	A single positive integer (equal number of units in every population) or a numeric vector of length `G` giving the number of subjects in each of the `G` populations, where `G` is the number of mean trajectories supplied through `fixed_coefficients`.
`target_times`	A numeric vector of the nominal (planned) measurement times, length M, one shared schedule for every unit. They need not be equally spaced. A degree-P model requires $M \geq P + 1$ occasions. Give either this or `time_range`.
`fixed_coefficients`	The population mean trajectory, as the coefficients of a polynomial in time, ordered from the intercept upward: `c(b0, b1, ..., bP)` encodes $b_0 + b_1 t + b_2 t^2 + \cdots + b_P t^P$. The length sets the polynomial order P (length 1 is order 0, a flat line at `b0`). For several populations, pass a *list* of equal-length coefficient vectors, one per population; the populations then differ in their mean trajectories but share the variance components below (the Raudenbush-Liu two-group setup).
`time_range`	Alternative to `target_times`: `c(lower,upper)` bounds from which each unit draws its own measurement times, so no two units share a schedule (e.g., age in weeks at testing rather than a fixed grade). Requires `occasions`; with `time_range`, the level-one error must be a single `error_variance` with the default independent structure, and `timing_sd` does not apply.
`occasions`	With `time_range`: a single positive integer (every unit measured the same number of times) or `c(min,max)`, from which each unit's number of measurement times is drawn uniformly. Every value must be at least $P + 1$ so each unit's trajectory identifies the polynomial.
`time_distribution`	Distribution of the unit-specific times over `time_range`; currently `"uniform"`.
`random_variances`	The between-subject variances of the polynomial coefficients (the diagonal of the level-two covariance matrix), as a single number recycled to all $P + 1$ coefficients or a vector of length $P + 1$. Default `0`, a fixed common trajectory with no between-subject heterogeneity. Entries may be zero coefficient by coefficient (e.g., a random intercept with a fixed slope).
`random_correlation`	Optional $(P + 1) \times (P + 1)$ correlation matrix among the random coefficients. Default `NULL` treats them as uncorrelated. Combined with `random_variances` it forms the level-two covariance $T = DRD$, with D the diagonal matrix of coefficient standard deviations.

`error_variance` The level-one (within-subject) measurement error variance σ_e^2. Most simply a single non-negative number (the same error variance at every occasion). It may instead be a length-M vector of per-occasion (heteroscedastic) error variances, or a full $M \times M$ error covariance matrix when the errors are correlated across occasions in an arbitrary (unstructured) way. For the common structured cases, give a scalar or vector here and set `error_structure` and `error_correlation` instead of building the matrix by hand. Specify exactly one of `error_variance` or `reliability`.

`reliability` A target measurement reliability in $(0, 1)$ from which σ_e^2 is derived (see Details). The value is the *average* per-occasion reliability across `target_times`; the per-occasion reliabilities, which generally differ from one another, are returned in the `"reliability_by_occasion"` attribute. Requires at least one positive entry in `random_variances`. The solved error variance is homoscedastic and may still be given an across-occasion correlation through `error_structure`. Specify exactly one of `error_variance` or `reliability`.

`error_structure`
The correlation pattern of the level-one errors across occasions: `"independent"` (the default, uncorrelated errors), `"ar1"` (a first-order autoregressive decay $\rho^{|j-k|}$, so errors at occasions closer in time are more alike), `"compound_symmetry"` (a constant correlation ρ between every pair of occasions), or `"toeplitz"` (a banded structure set by the lag-1 through lag-$(M-1)$ correlations). Ignored when `error_variance` is a full covariance matrix, which already fixes the structure.

`error_correlation`
The correlation parameter(s) for `error_structure`: a single number ρ for `"ar1"` and `"compound_symmetry"`, or a vector of the lag-1 to lag-$(M-1)$ correlations for `"toeplitz"`. Left `NULL` for `"independent"`. The implied correlation matrix must be positive semidefinite.

`timing_sd` The standard deviation of the difference between a subject's actual and nominal assessment time, as a single number recycled to all occasions or a vector of length M. Default `0`, every subject measured exactly on schedule. A positive value draws each subject's actual time at occasion m as $\tau_m + N(0, \text{timing\_sd}_m^2)$ and evaluates that subject's true score at the actual time, while the nominal target is retained in a separate column (see Details).

Details

The model. Subject i in population g has a random coefficient vector $\pi_i = (\pi_{i0}, \ldots, \pi_{iP})$ drawn from a multivariate normal with mean the population's `fixed_coefficients` β_g and covariance T. The latent trajectory is the polynomial $\mu_i(t) = \sum_{k=0}^{P} \pi_{ik}\, t^k$, and the observed score at a measurement time t is $y = \mu_i(t) + e$, with $e \sim N(0, \sigma_e^2)$ independent across occasions. Order 0 collapses to a flat line $\mu_i(t) = \pi_{i0}$; order 1 is the straight-line growth model underlying Raudenbush and Liu (2001) and `ss_power_pcm`.

Coefficient metric. The coefficients here are the ordinary (raw) polynomial coefficients on t^k, which is the most transparent metric for specifying a trajectory. The derivative-scaled change coefficient used by `ss_power_pcm` and the Raudenbush-Liu power formulas is $P!$ times the leading (highest-order) coefficient supplied here, so a quadratic with `fixed_coefficients = c(b0, b1, b2)` corresponds to a Raudenbush-Liu quadratic change coefficient of $2!\, b_2 = 2b_2$.

Reliability varies by occasion. At a measurement time t the implied between-subject (true-score) variance is the quadratic form $c(t)^\top T\, c(t)$ with $c(t) = (1, t, t^2, \ldots, t^P)^\top$, so the classical reliability of the observed score,

$$\rho_{XX}(t) = \frac{c(t)^\top T\, c(t)}{c(t)^\top T\, c(t) + \sigma_e^2},$$

generally *changes from occasion to occasion*: a growth measurement is not equally reliable everywhere, because the spread of true scores depends on where in time you measure relative to the centering of the polynomial and the random-effect structure. When `reliability` is supplied, σ_e^2 is solved (by `uniroot`) so that the average of $\rho_{XX}(t)$ over the nominal `target_times` equals the requested value; the occasion-by-occasion reliabilities are returned in the `"reliability_by_occasion"` attribute so the variation is visible rather than hidden behind a single number. Reliability is only meaningful when there is true-score variance to detect, so this route requires `random_variances` to be positive for at least one coefficient.

Measurement errors need not be independent or equal. The simplest model adds an independent, equal-variance error at every occasion, but repeated measurements of the same person are often correlated (an unmodeled state, a rater, or an instrument carries over from one wave to the next) and may be more or less variable at different waves. The level-one errors are drawn from $N(0, \Sigma_e)$, and Σ_e can be set three ways: a scalar or per-occasion `error_variance` combined with an `error_structure` (`"ar1"` for autoregressive decay, the natural choice when occasions are ordered in time; `"compound_symmetry"` for an equicorrelated error; `"toeplitz"` for a general banded pattern), or a full covariance matrix passed directly as `error_variance`. Because classical reliability at an occasion is a marginal quantity, it depends only on the diagonal of Σ_e; the across-occasion error correlation leaves `"reliability_by_occasion"` unchanged but does affect how a mixed model that assumes independent errors performs, which is exactly the kind of misspecification this simulator is meant to let a user study.

Assessment timing is rarely exact. Designs are written as if every subject is measured at the same fixed times ("the 7-day follow-up"), but in practice people arrive early or late, so the actual time differs from the nominal target. Setting `timing_sd > 0` draws each subject's actual time per occasion and evaluates the true score *at the time the measurement really happened*, while `target_time` keeps the nominal value an analyst would typically use. Analyzing on the nominal time when the data were in fact collected on jittered times biases estimates of the change coefficients, and the bias grows with the order of the trend and with the size of the timing variability. The two time columns let a user quantify that bias by fitting the same model on `time` versus `target_time`.

Why the closed-form Raudenbush-Liu planner does not cover all of this. The power formulas in Raudenbush and Liu (2001), carried by `ss_power_pcm`, are exact under three assumptions this simulator can relax: every subject is measured at the *same*, equally spaced, *error-free* occasion times; the level-one error variance is a single constant (so the closed form needs no notion of an occasion-varying reliability); and the within-subject sampling variance of the change coefficient has the known form $V = \sigma_e^2 f^{2p}(M-p-1)!/[K_p(M+p)!]$. Those assumptions buy a clean formula, but real designs violate them: assessments drift in time, and reliability is not the same at every wave. This function is the Monte Carlo complement that lets a researcher generate data under the messier reality and check how far the closed-form power and the fitted estimates can be trusted.

Value

A long-format `data.frame` with one row per subject-occasion and the columns

`id` Factor uniquely identifying each subject.

`population` Factor with `G` levels (`"1"`, ...) giving each unit's population (its data generating parameter vector). With one parameter vector there is one level.

`occasion` Integer occasion index, `1` to M.

`target_time` The nominal (planned) measurement time.

`time` The actual measurement time (equal to `target_time` when `timing_sd = 0`, otherwise jittered).

`true_score` The subject's latent trajectory value at the actual time, before level-one error.

`y` The observed score, `true_score` plus level-one error.

The returned object carries attributes `"error_variance"` (the σ_e^2 used, a scalar when the errors are homoscedastic and independent, otherwise the vector of per-occasion error variances), `"error_covariance"` (the full $M \times M$ level-one error covariance actually used), `"reliability_by_occasion"` (the per-occasion reliabilities at the nominal times), `"random_covariance"` (the level-two covariance T), `"polynomial_order"` (P), and `"schedule"` (`"shared"` or `"unit_specific"`). With `time_range` there is no shared occasion grid, so `"error_covariance"` is `NA` and `"reliability_by_occasion"` is `NA`. The format is directly usable with `plot_trajectories` and with mixed-model fitters such as `nlme::lme()` or `lme4::lmer()`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Rausch, J. R. (2011). Sample size planning for longitudinal models: Accuracy in parameter estimation for polynomial change parameters. *Psychological Methods, 16*(4), 391–405. doi:10.1037/a0023352

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 15 on the analysis of repeated measures and growth.)

Raudenbush, S. W., & Liu, X.-F. (2001). Effects of study duration, frequency of observation, and sample size on power in studies of group differences in polynomial change. *Psychological Methods, 6*(4), 387–401. doi:10.1037/1082989X.6.4.387

See Also

`ss_power_pcm` for closed-form power planning on the same model, `plot_trajectories` to visualize the simulated curves, and `mvrnorm` for the random-coefficient draw.

Other data simulators: `simulate_ancova_data()`, `simulate_ancova_factorial_data()`, `simulate_anova_data()`, `simulate_longitudinal_gompertz()`, `simulate_longitudinal_logistic()`, `simulate_longitudinal_negative_exp`
`simulate_longitudinal_richards()`, `simulate_regression_data()`

Examples

```
# 1. One population of linear growers with a random intercept and a random slope,
#    measured yearly for four years (five occasions), with the level-one
#    error set directly.
```

```
set.seed(113)
d <- simulate_longitudinal_polynomial(
  n                 = 50,
  target_times      = 0:4,
  fixed_coefficients = c(10, 1.5),          # intercept 10, slope 1.5 per year
  random_variances   = c(4, 0.25),          # var(intercept) = 4, var(slope) = .25
  error_variance     = 1
)
head(d)

# 2. Two populations that differ only in their slope (a treatment that changes
#    the rate of growth). Pass a list of coefficient vectors, one per population.
set.seed(113)
two <- simulate_longitudinal_polynomial(
  n                 = c(40, 40),
  target_times      = 0:4,
  fixed_coefficients = list(control = c(10, 1.0), treatment = c(10, 1.8)),
  random_variances   = c(4, 0.25),
  error_variance     = 1
)
aggregate(y ~ population + occasion, data = two, FUN = mean)

# 3. Pin the level-one error to a target reliability instead of setting it
#    directly. The single number is the average reliability across occasions;
#    the per-occasion values differ and are returned as an attribute.
set.seed(113)
rel <- simulate_longitudinal_polynomial(
  n                 = 100,
  target_times      = 0:4,
  fixed_coefficients = c(10, 1.5),
  random_variances   = c(4, 0.25),
  reliability        = 0.80
)
attr(rel, "error_variance")
round(attr(rel, "reliability_by_occasion"), 3)   # not constant across waves

# 4. Assessment-time jitter: the nominal "yearly" schedule, but subjects
#    actually arrive a little early or late (SD of about six weeks on a
#    one-year scale). The nominal and actual times are kept in separate
#    columns so the consequences of analyzing on the nominal time can be
#    studied.
set.seed(113)
jit <- simulate_longitudinal_polynomial(
  n                 = 30,
  target_times      = 0:4,
  fixed_coefficients = c(10, 1.5),
  random_variances   = c(4, 0.25),
  error_variance     = 1,
  timing_sd          = 0.12
)
head(jit[, c("id", "occasion", "target_time", "time")])

# 4b. Unit-specific measurement times: each of 12 children is tested
```

```
#     between 40 and 90 weeks of age, five to nine times, no two on the
#     same schedule. The level-one error is a single variance; the
#     "schedule" attribute records the design.
set.seed(113)
ages <- simulate_longitudinal_polynomial(
  n                 = 12,
  time_range        = c(40, 90),
  occasions         = c(5, 9),
  fixed_coefficients = c(10, 0.5),
  random_variances   = c(4, 0.01),
  error_variance     = 2
)
attr(ages, "schedule")
table(table(ages$id))   # units per occasion count

# 5. A flat line (order 0): no growth, only a random subject level and
#    measurement error. The coefficient vector has length one.
set.seed(113)
flat <- simulate_longitudinal_polynomial(
  n                 = 20,
  target_times      = 0:4,
  fixed_coefficients = 5,
  random_variances   = 2,
  error_variance     = 1
)
head(flat)

# 6. Autocorrelated measurement error: the same error variance at each wave,
#    but the level-one errors decay as an AR(1) process (errors at adjacent
#    occasions correlate 0.5), the kind of dependence a model assuming
#    independent errors would miss. The full error covariance is returned.
set.seed(113)
ar <- simulate_longitudinal_polynomial(
  n                 = 40,
  target_times      = 0:4,
  fixed_coefficients = c(10, 1.5),
  random_variances   = c(4, 0.25),
  error_variance     = 1,
  error_structure    = "ar1",
  error_correlation  = 0.5
)
round(attr(ar, "error_covariance"), 3)
```

---

simulate_longitudinal_richards

Simulate Data From a Richards Change (Growth) Model

Description

Generates longitudinal data from a random-coefficients Richards change model, the flexible sigmoidal family whose point of inflection is itself a parameter rather than a fixed fraction of total change (Richards, 1959). The deterministic part of the five parameter Richards curve is

$$\mu(t) = \frac{\alpha}{\left(1 + \delta \exp(-\gamma(t - \beta))\right)^{1/\delta}} + \zeta,$$

the parameterization of Kelley (2005, 2008). The Richards family subsumes the package's other sigmoidal curves as special cases: at $\delta = 1$ it is exactly the logistic, and in the limit $\delta \to 0$ it is the Gompertz, so δ lets the data, or the theory, choose where in its course the process turns from acceleration to deceleration. Guo, Cheng, and Kelley (2016) use this flexibility to model self-replicating malware propagation, where the network structure moves the inflection of the outbreak.

Usage

```
simulate_longitudinal_richards(
  n,
  target_times = NULL,
  fixed_parameters,
  time_range = NULL,
  occasions = NULL,
  time_distribution = "uniform",
  random_variances = 0,
  random_correlation = NULL,
  error_variance = NULL,
  reliability = NULL,
  error_structure = c("independent", "ar1", "compound_symmetry", "toeplitz"),
  error_correlation = NULL,
  timing_sd = 0
)
```

Arguments

`n`	A single positive integer, the number of units (persons, animals, trees, classrooms) whose trajectories are drawn, or a vector giving the number of units for each population, one entry per parameter vector in `fixed_parameters`.
`target_times`	Numeric vector of the nominal measurement times, one shared schedule for every unit. Give either this or `time_range`.
`fixed_parameters`	The population parameters `c(alpha = , beta = , gamma = , delta = , zeta = )` (an unnamed length-5 vector is taken in that order), or a list of such vectors, one per population (distinct data generating parameter vectors):

`alpha` The total change: the curve travels `alpha` units from the lower asymptote `zeta` to the upper asymptote $\alpha + \zeta$ (for $\gamma > 0$).

`beta` The point of inflection on the time axis.

`gamma` The curvature: how sharply the curve rises through its inflection. Negative values flip the curve to decreasing.

`delta` The shape parameter ($\delta > 0$): where the inflection falls on the outcome axis, $y^* = \alpha(1+\delta)^{-1/\delta} + \zeta$. At $\delta = 1$ the inflection sits at half the total change (the logistic); as $\delta \to 0$ it slides down to $\alpha/e + \zeta$, about 36.8% (the Gompertz); larger δ pushes it later than halfway.

`zeta` The lower asymptote (for $\gamma > 0$). The intercept is $\phi = \alpha(1+\delta\exp(\gamma\beta))^{-1/\delta} + \zeta$.

`time_range` Alternative to `target_times`: `c(lower,upper)` bounds from which each unit draws its own measurement times, so no two units share a schedule (e.g., age in weeks at testing rather than a fixed grade). Requires `occasions`; with `time_range`, the level-one error must be a single `error_variance` with the default independent structure, and `timing_sd` does not apply.

`occasions` With `time_range`: a single positive integer (every unit measured the same number of times) or `c(min,max)`, from which each unit's number of measurement times is drawn uniformly.

`time_distribution`
Distribution of the unit-specific times over `time_range`; currently `"uniform"`.

`random_variances`
Between-unit variances of `(alpha, beta, gamma, delta, zeta)`, a single number recycled to all five or a length-5 vector. Default `0`. A named vector over any subset of the parameter names (e.g., `c(delta = 0.04)`) varies only those named and leaves the rest fixed. A positive variance on `delta` must be small enough that no unit's drawn δ falls at or below zero, where the curve is undefined; the function stops with a count if that happens.

`random_correlation`
Optional 5-by-5 correlation matrix among the random parameters; default uncorrelated.

`error_variance`, `reliability`, `error_structure`, `error_correlation`, `timing_sd`
The level-one error and assessment-time machinery, with the same meaning as in `simulate_longitudinal_polynomial`: specify exactly one of `error_variance` or `reliability` (solved here through the first-order delta method true-score variance; because the curve is nonlinear in its parameters, that approximation, and the `reliability_by_occasion` attribute with it, can drift from the realized variance ratio when the random variances are large relative to the mean curve); `error_structure` and `error_correlation` set the across-occasion error correlation; `timing_sd` jitters the actual assessment times.

Details

The logistic and the Gompertz fix where the inflection falls as a fraction of total change (50% and 36.8%); choosing between them is choosing that fraction by assumption. The Richards curve makes the fraction estimable through δ, at the price of one more parameter and a harder estimation problem, since δ and γ carry overlapping information in finite samples (Richards, 1959; Kelley, 2005). Simulating from the Richards family at several δ values is the natural way to study whether a design can tell those shapes apart.

Value

A long-format `data.frame` with columns `id`, `population`, `occasion`, `target_time`, `time`, `true_score`, and `y`, directly usable with `plot_trajectories` and nonlinear mixed-model fitters such as `nlme::nlme()`. Attributes carry the `model`, `fixed_parameters`, `random_covariance`, `error_variance`, `error_covariance`, `reliability_by_occasion`, and `schedule` (`"shared"` or `"unit_specific"`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Guo, H., Cheng, H. K., & Kelley, K. (2016). Impact of network structure on malware propagation: A growth curve perspective. *Journal of Management Information Systems, 33*(1), 296–325.

Kelley, K. (2005). *Estimating nonlinear change models in heterogeneous populations when class membership is unknown: Defining and developing the latent classification differential change model* (Doctoral dissertation). University of Notre Dame.

Kelley, K. (2008). Nonlinear change models in populations with unobserved heterogeneity. *Methodology, 4*(3), 97–112.

Richards, F. J. (1959). A flexible growth function for empirical use. *Journal of Experimental Botany, 10*(2), 290–301.

See Also

`simulate_longitudinal_logistic` (the $\delta = 1$ special case), `simulate_longitudinal_gompertz` (the $\delta \to 0$ limit), `simulate_longitudinal_negative_exponential`, `simulate_longitudinal_polynomial`, `plot_trajectories`.

Other data simulators: `simulate_ancova_data()`, `simulate_ancova_factorial_data()`, `simulate_anova_data()`, `simulate_longitudinal_gompertz()`, `simulate_longitudinal_logistic()`, `simulate_longitudinal_negative_ex`
`simulate_longitudinal_polynomial()`, `simulate_regression_data()`

Examples

```
# The family in one panel: five Richards curves sharing alpha, beta,
# gamma, and zeta and differing only in the shape parameter delta.
# delta near 0 is the Gompertz, delta = 1 is the logistic, and
# larger delta pushes the inflection later than halfway. Each curve
# is its own population of size one, so a single call draws the whole panel.
panel <- simulate_longitudinal_richards(
  n = 1, target_times = seq(0, 12, by = 0.1),
  fixed_parameters = list(
    c(alpha = 80, beta = 6, gamma = 0.9, delta = 0.02, zeta = 10),
    c(alpha = 80, beta = 6, gamma = 0.9, delta = 0.25, zeta = 10),
    c(alpha = 80, beta = 6, gamma = 0.9, delta = 1.00, zeta = 10),
    c(alpha = 80, beta = 6, gamma = 0.9, delta = 3.00, zeta = 10),
    c(alpha = 80, beta = 6, gamma = 0.9, delta = 8.00, zeta = 10)),
  error_variance = 0
)
plot_trajectories(panel, id = "id", time = "time",
```

```
                  outcome = "true_score", group = "population")

# Individual differences in a single parameter: only the shape
# varies (a named entry leaves the other variances at zero), so the
# curves agree on floor, ceiling, timing, and curvature yet turn
# from acceleration to deceleration at different heights.
set.seed(113)
d_delta <- simulate_longitudinal_richards(
  n = 25, target_times = seq(0, 12, by = 0.5),
  fixed_parameters = c(alpha = 80, beta = 6, gamma = 0.9,
                       delta = 1, zeta = 10),
  random_variances = c(delta = 0.04), error_variance = 0
)
plot_trajectories(d_delta, id = "id", time = "time",
                  outcome = "true_score")

# Individual differences in every parameter, plus level-one error:
# a late-inflecting outbreak-style curve, with delta = 3 placing the
# inflection at about 63% of total change, (1 + 3)^(-1/3).
set.seed(113)
d <- simulate_longitudinal_richards(
  n = 30, target_times = 0:12,
  fixed_parameters = c(alpha = 80, beta = 6, gamma = 0.9,
                       delta = 3, zeta = 10),
  random_variances = c(alpha = 36, beta = 1, gamma = 0.01,
                       delta = 0.04, zeta = 9),
  error_variance = 16
)
plot_trajectories(d, id = "id", time = "time", outcome = "y")

# delta = 1 reproduces the logistic exactly: with no randomness and
# no error, the two simulators return identical true scores.
d_r <- simulate_longitudinal_richards(
  n = 1, target_times = 0:5,
  fixed_parameters = c(alpha = 80, beta = 3, gamma = 1,
                       delta = 1, zeta = 10),
  error_variance = 0)
d_l <- simulate_longitudinal_logistic(
  n = 1, target_times = 0:5,
  fixed_parameters = c(alpha = 80, beta = 3, gamma = 1, zeta = 10),
  error_variance = 0)
all.equal(d_r$true_score, d_l$true_score)
```

---

simulate_regression_data

Simulate Data From a Multivariate Normal Multiple-Regression Model

---

Description

Generates random data $(Y, X_1, \ldots, X_p)$ jointly multivariate normal with user-specified marginal means, marginal SDs, and full correlation structure. Useful as a backbone for sensitivity analyses, Monte Carlo studies of regression sample size methods, and pedagogical demonstrations.

Usage

```
simulate_regression_data(
  N,
  p,
  rho_YX,
  rho_XX = NULL,
  mu_Y = 0,
  mu_X = 0,
  sigma_Y = 1,
  sigma_X = 1,
  seed = NULL,
  column_names = NULL
)
```

Arguments

N	The total sample size (a positive integer $\geq p + 2$).
p	The number of predictor variables.
rho_YX	A numeric vector of length p giving the population correlations between Y and each predictor X_j.
rho_XX	A $p \times p$ symmetric correlation matrix for the predictors. Defaults to the identity matrix (orthogonal predictors).
mu_Y	The population mean of Y (default 0).
mu_X	A numeric vector of length p giving the population mean of each predictor (default 0, recycled across predictors).
sigma_Y	The population standard deviation of Y (default 1, in which case the simulated Y is on the standardized scale).
sigma_X	A single number or a numeric vector of length p giving the population standard deviation of each predictor (default 1, standardized).
seed	Optional integer random seed for reproducibility (default NULL).
column_names	Optional character vector of length p + 1 giving column names for the returned data.frame; defaults to c("y", "x1", "x2", ..., "xp").

Details

Internally the joint correlation matrix is assembled as

$$R = \begin{pmatrix} 1 & \rho_{YX}^{\top} \\ \rho_{YX} & R_{XX} \end{pmatrix},$$

converted to a covariance matrix via the supplied SDs, and N draws are taken using mvrnorm. The resulting Y and predictors satisfy the requested marginal means and standard deviations and (in expectation) the requested correlation structure.

Value

A data.frame with N rows and p + 1 columns: the outcome Y (first column) followed by predictors $X_1, \ldots, X_p$.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

simulate_ancova_data, simulate_anova_data, ss_aipe_R2, ss_aipe_reg_coef, ci_R2

Other data simulators: simulate_ancova_data(), simulate_ancova_factorial_data(), simulate_anova_data(), simulate_longitudinal_gompertz(), simulate_longitudinal_logistic(), simulate_longitudinal_negative_exp simulate_longitudinal_polynomial(), simulate_longitudinal_richards()

Examples

```
# Five orthogonal predictors, each correlating .30 with Y.
set.seed(113)
d <- simulate_regression_data(
  N      = 200,
  p      = 5,
  rho_YX = rep(0.30, 5)
)
summary(lm(y ~ ., data = d))$r.squared   # about 0.45: five predictors, each 0.30^2

# Predictors with shared structure (exchangeable correlation matrix).
rho_XX <- matrix(0.5, nrow = 5, ncol = 5); diag(rho_XX) <- 1
simulate_regression_data(
  N      = 300,
  p      = 5,
  rho_YX = c(.50, .40, .30, .20, .10),
  rho_XX = rho_XX,
  seed   = 113
)[1:3, ]
```

---

skewness *Bias-Corrected Sample Skewness*

Description

Computes the sample skewness of a numeric vector using the bias-corrected (SAS/SPSS Type 2) formula. Skewness measures asymmetry of the distribution: zero is symmetric, positive values indicate a right-tail heavier than the left, negative values the reverse.

Usage

```
skewness(x, na_rm = TRUE)
```

Arguments

x	A numeric vector.
na_rm	Logical. If TRUE (the default), missing values are removed before computation. If FALSE, the result is NA when x contains any NA.

Details

The reported value is

$$\hat{\gamma}_1^{(2)} = \frac{n}{(n-1)(n-2)} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s} \right)^3,$$

where s is the (divisor-$n-1$) sample standard deviation. This is sometimes called the "Type 2" or SAS/SPSS-default form; it is approximately unbiased under normality.

Why isn't this in base R? R Core has historically deferred higher-order moment statistics to contributed packages, in part because three popular formulas exist (biased Type 1, bias-corrected Type 2, and Minitab Type 3) and choosing a default would be opinionated. DMAR adopts Type 2, which is the form most often used in psychometric reporting and the one already used internally by descriptives.

Diagnostic interpretation. As a rough rule of thumb, $|\text{skewness}| > 2$ is sometimes flagged as indicative of departures from normality large enough to threaten normal-theory inference (e.g., maximum likelihood estimation in factor analysis or structural equation modeling).

Value

A single numeric value: the bias-corrected sample skewness, or NA_real_ when fewer than three non-missing observations are available or when the sample standard deviation is zero.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Joanes, D. N., & Gill, C. A. (1998). Comparing measures of sample skewness and kurtosis. *The Statistician, 47*(1), 183–189. doi:10.1111/14679884.00122

See Also

kurtosis, descriptives

Other descriptive statistics: descriptives(), kurtosis()

Examples

```
# Symmetric data: skewness near zero.
set.seed(113)
skewness(rnorm(1000))

# Right-skewed data: positive value.
skewness(rexp(1000, rate = 1))

# The classic 1:5 example: exactly symmetric (returns 0).
skewness(1:5)
```

---

smd *Standardized Mean Difference*

Description

Estimates the standardized mean difference (Cohen's *d*), the difference between two group means divided by the pooled standard deviation, from either raw data or summary statistics. Expressing the difference in standard deviation units frees the comparison from the raw measurement units, so effects can be compared across measures and studies; either the ordinary or the unbiased (Hedges, 1981) estimate can be returned.

Usage

```
smd(
  group_1 = NULL,
  group_2 = NULL,
  mean_1 = NULL,
  mean_2 = NULL,
  s_1 = NULL,
  s_2 = NULL,
  s = NULL,
  n_1 = NULL,
  n_2 = NULL,
  unbiased = FALSE
)
```

Arguments

group_1	Raw data for group 1
group_2	Raw data for group 2
mean_1	The mean of group 1
mean_2	The mean of group 2
s_1	The standard deviation of group 1 (i.e., the square root of the unbiased estimator of the population variance)

s_2	The standard deviation of group 2 (i.e., the square root of the unbiased estimator of the population variance)
s	The pooled group standard deviation (i.e., the square root of the unbiased estimator of the population variance)
n_1	The sample size within group 1
n_2	The sample size within group 2
unbiased	Returns the unbiased estimate of the standardized mean difference

Details

When `unbiased=TRUE`, the unbiased estimate of the standardized mean difference is returned (Hedges, 1981).

Value

A 1-row `data.frame` with columns `term` (`"smd"`) and `value` (the estimated standardized mean difference).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*(4), 532–574. doi:10.1177/0013164401614002

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Kelley, K. (2005) The effects of nonnormal distributions on confidence intervals around the standardized mean difference: Bootstrap and parametric confidence intervals, *Educational and Psychological Measurement, 65*, 51–69. doi:10.1177/0013164404264850

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons and Chapter 3 on one-way ANOVA.)

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

smd_c, ci_smd, ci_smd_c, ss_aipe_smd, ss_power_smd, plot_smd, ci_nc_t

Examples

```
# Generate sample data.
set.seed(113)
g.1 <- rnorm(n = 25, mean = .5, sd = 1)
g.2 <- rnorm(n = 25, mean = 0, sd = 1)
smd(group_1 = g.1, group_2 = g.2)

M.x <- .66745
M.y <- .24878
sd <- 1.048
smd(mean_1 = M.x, mean_2 = M.y, s = sd)

M.x <- .66745
M.y <- .24878
n1 <- 25
n2 <- 25
sd.1 <- .95817
sd.2 <- 1.1311
smd(mean_1 = M.x, mean_2 = M.y, s_1 = sd.1, s_2 = sd.2, n_1 = n1, n_2 = n2)

smd(mean_1 = M.x, mean_2 = M.y, s_1 = sd.1, s_2 = sd.2, n_1 = n1, n_2 = n2,
    unbiased = TRUE)
```

---

smd_c *Standardized Mean Difference Using the Control Group as the Basis of Standardization*

Description

Estimates the standardized mean difference using the control group standard deviation as the basis of standardization (Glass's *g*), from either raw data or summary statistics, in ordinary or unbiased form. Standardizing by the control group alone keeps the scale of the effect free of any treatment effect on variability.

Usage

```
smd_c(
  group_T = NULL,
  group_C = NULL,
  mean_T = NULL,
  mean_C = NULL,
  s_C = NULL,
  n_C = NULL,
  unbiased = FALSE
)
```

Arguments

group_T	Raw data for the treatment group
group_C	Raw data for the control group
mean_T	The mean of the treatment group
mean_C	The mean of the control group
s_C	The standard deviation of the control group (i.e., the square root of the unbiased estimator of the population variance)
n_C	The sample size of the control group
unbiased	Returns the unbiased estimate of the standardized mean difference using the standard deviation of the control group

Details

When unbiased=TRUE, the unbiased estimate of the standardized mean difference (using the control group as the basis of standardization) is returned (Hedges, 1981). Although the unbiased estimate of the standardized mean difference is not often reported, at least at the present time, it is nevertheless made available to those who are interested in calculating this quantity.

Value

A 1-row data.frame with columns term ("smd_c") and value (the estimated standardized mean difference using the control group standard deviation as the basis of standardization).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Glass, G. V. (1976). Primary, secondary, and meta-analysis of research. *Educational Researcher, 5*, 3–8.

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons and Chapter 3 on one-way ANOVA.)

See Also

smd, ci_nc_t

Examples

```
# Generate sample data.
set.seed(113)
g.T <- rnorm(n = 25, mean = .5, sd = 1)
g.C <- rnorm(n = 25, mean = 0, sd = 1)
smd_c(group_T = g.T, group_C = g.C)

M.T <- .66745
M.C <- .24878
sd.c <- 1.1311
n.c <- 25
smd_c(mean_T = M.T, mean_C = M.C, s_C = sd.c)
smd_c(mean_T = M.T, mean_C = M.C, s_C = sd.c, n_C = n.c, unbiased = TRUE)
```

---

smd_trimmed *Robust Standardized Mean Difference (Algina-Keselman-Penfield)*

Description

Computes the Algina, Keselman, and Penfield (2005) robust standardized mean difference, which replaces the sample means and pooled SD in Cohen's d with their trimmed-mean and Winsorized-SD counterparts:

$$d_R \;=\; 0.642 \cdot \frac{\bar{X}_{t,\,1} - \bar{X}_{t,\,2}}{s_{W,\,p}},$$

where $\bar{X}_{t,\,j}$ is the trimmed mean of group j, $s_{W,\,p}$ is the pooled Winsorized standard deviation, and 0.642 is the Algina-Keselman-Penfield (2005) constant chosen so that d_R equals Cohen's δ when the data are normal. Returns the point estimate, a noncentral *t* confidence interval, and the trimmed / Winsorized summary statistics.

Usage

```
smd_trimmed(x, y, trim = 0.2, conf_level = 0.95)
```

Arguments

`x, y`	Numeric vectors of observations from the two groups.
`trim`	Proportion to trim from each tail (and Winsorize from each tail). Must be in $[0, 0.5)$. Default `0.20` (Wilcox's 2017 recommended setting).
`conf_level`	Confidence level for the CI. Default `0.95`.

Details

Why robust. Under heavy-tailed or skewed marginal distributions, the conventional Cohen's *d* has very large standard error and biased coverage. Kelley (2005) documents the coverage distortion of parametric confidence intervals for the standardized mean difference under nonnormal distributions. Replacing means by 20%- trimmed means and SD by 20%-Winsorized SD yields an estimator

whose efficiency under normality is roughly 96% (Wilcox, 2017, ch. 5) and whose efficiency under heavy-tailed contamination is substantially higher than Cohen's *d*.

The 0.642 constant. $0.642 = \mathrm{SD}(X_W)/\mathrm{SD}(X) = \sqrt{\mathrm{Var}(X_W)/\mathrm{Var}(X)}$ when $X \sim N(0,1)$ and X_W is the 20%-Winsorized version. Choosing this constant makes $d_R = \delta$ when the data are normal, so the new estimator is on the same scale as Cohen's *d*.

CI. The CI follows the construction of Keselman, Algina, Lix, Wilcox, and Deering (2008): Yuen's (1974) *t*-statistic on the trimmed-mean difference (their Equation 8) is referred to a noncentral *t* distribution with the Yuen-Welch approximate degrees of freedom (their Equation 9), the noncentrality parameters whose tail probabilities bracket the observed statistic are located with `ci_nc_t`, and those limits are rescaled to the d_R metric. The degrees of freedom are reported in the `df_yuen` row of the returned table. At `trim = 0` the construction reduces to the Welch approximate degrees of freedom interval; for the exact equal-variance interval on the untrimmed standardized mean difference use `ci_smd`.

Value

A `data.frame` with rows for the robust *d* estimate, the lower/upper CI bounds, the per-group trimmed means, the per-group Winsorized SDs, the pooled Winsorized SD, and the effective sample sizes (after trimming).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Algina, J., Keselman, H. J., & Penfield, R. D. (2005). An alternative to Cohen's standardized mean difference effect size: A robust parameter and confidence interval in the two independent groups case. *Psychological Methods, 10*(3), 317–328. doi:10.1037/1082989X.10.3.317

Kelley, K. (2005). The effects of nonnormal distributions on confidence intervals around the standardized mean difference: Bootstrap and parametric confidence intervals. *Educational and Psychological Measurement, 65*(1), 51–69. doi:10.1177/0013164404264850

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Keselman, H. J., Algina, J., Lix, L. M., Wilcox, R. R., & Deering, K. N. (2008). A generally robust approach for testing hypotheses and setting confidence intervals for effect sizes. *Psychological Methods, 13*(2), 110–129. doi:10.1037/1082989X.13.2.110

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons and Chapter 3 on one-way ANOVA.)

Wilcox, R. R. (2017). *Introduction to robust estimation and hypothesis testing* (4th ed.). Academic Press.

Yuen, K. K. (1974). The two-sample trimmed *t* for unequal population variances. *Biometrika, 61*(1), 165–170.

See Also

smd, var_smd_trimmed, ci_smd, ci_nc_t

Other effect size estimates: cles(), cliff_delta(), correction_for_attenuation(), eta_squared(), eta_squared_generalized(), eta_squared_partial(), expected_partial_r(), expected_r(), expected_smd(), nnt_from_smd(), omega_squared(), omega_squared_partial(), probability_of_superiority_pai proportion_of_superiority(), responder_analysis()

Examples

```
# 1. Two normal groups: robust d agrees closely with Cohen's d.
set.seed(113)
x <- rnorm(40, 0,   1); y <- rnorm(40, 0.5, 1)
smd_trimmed(x, y)

# 2. Contaminated y: a few outliers; robust d shifts much less
#        than Cohen's d.
set.seed(113)
x <- rnorm(40, 0, 1)
y <- c(rnorm(38, 0.5, 1), 30, -25)
smd_trimmed(x, y)
smd(x, y)
```

---

ss_aipe_c *Sample Size Planning for an ANOVA Contrast From the Accuracy in Parameter Estimation (AIPE) Perspective*

Description

Plans the sample size *per group* so that the confidence interval for an unstandardized contrast of means in a fixed effects analysis of variance is sufficiently narrow, following the accuracy in parameter estimation (AIPE) approach: the design goal is a contrast estimated with the precision the research question requires, not merely one detected as nonzero. AIPE sample size planning for ANOVA and ANCOVA contrasts is developed in Lai and Kelley (2012).

Usage

```
ss_aipe_c(
  error_variance = NULL,
  c_weights,
  width,
  conf_level = 0.95,
  assurance = NULL,
  MSwithin = NULL,
  SD = NULL,
  ...
)
```

Arguments

error_variance	The common error variance; i.e., the mean square error
c_weights	The contrast weights
width	The desired full width of the obtained confidence interval
conf_level	The desired confidence interval coverage, (i.e., 1 - Type I error rate)
assurance	Parameter to ensure that the obtained confidence interval width is narrower than the desired width with a specified degree of certainty (must be NULL or between zero and unity)
MSwithin	An alias for error_variance
SD	The standard deviation of the common error in ANOVA model
...	Allows one to potentially include parameter values for inner functions

Value

A 1-row data.frame with columns term and value:

necessary_n_per_group
: the necessary sample size *per group*

Note

Be sure to use the error variance and not its square root (i.e., the standard deviation of the errors).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., Maxwell, S. E., & Rausch, J. R. (2003). Obtaining power or obtaining precision: Delineating methods of sample size planning. *Evaluation and the Health Professions, 26*(3), 258–287. doi:10.1177/0163278703255242

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

ss_aipe_sc, ss_aipe_c_ancova, ci_c

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Suppose the population error variance of some three-group ANOVA model
# is believed to be 40. The researcher is interested in the difference
# between the mean of group 1 and the average of means of group 2 and 3.
# To plan the sample size so that, with 90 percent certainty, the
# obtained 95 percent full confidence interval width is no wider than 3:

ss_aipe_c(error_variance = 40, c_weights = c(1, -0.5, -0.5),
          width = 3, assurance = .90)
```

---

ss_aipe_c_ancova *Sample Size Planning for a Contrast in Randomized ANCOVA From the Accuracy in Parameter Estimation (AIPE) Perspective*

Description

Plans the sample size per group so that the confidence interval for an unstandardized contrast in a one-covariate randomized ANCOVA is sufficiently narrow, following the accuracy in parameter estimation (AIPE) approach. To the extent the covariate correlates with the response, the covariate adjustment shrinks the error variance, so the desired precision is reached with a smaller sample size than the corresponding ANOVA design requires.

Usage

```
ss_aipe_c_ancova(
  error_var_ancova = NULL,
  error_var_anova = NULL,
  rho = NULL,
  c_weights,
  width,
  conf_level = 0.95,
  assurance = NULL
)
```

Arguments

error_var_ancova	The population error variance of the ANCOVA model (i.e., the mean square within of the ANCOVA model)
error_var_anova	The population error variance of the ANOVA model (i.e., the mean square within of the ANOVA model)
rho	The population correlation coefficient of the response and the covariate
c_weights	The contrast weights
width	The desired full width of the obtained confidence interval

`conf_level` The desired confidence interval coverage, (i.e., 1 - Type I error rate)

`assurance` Parameter to ensure that the obtained confidence interval width is narrower than the desired width with a specified degree of certainty (must be NULL or between zero and unity)

Details

Either the error variance of the ANCOVA model or of the ANOVA model can be used to plan the appropriate sample size per group. When using the error variance of the ANOVA model to plan sample size, the correlation coefficient of the response and the covariate is also needed.

Value

A 1-row `data.frame` with columns `term` and `value`:

`necessary_n_per_group`
The necessary sample size *per group*

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., Maxwell, S. E., & Rausch, J. R. (2003). Obtaining power or obtaining precision: Delineating methods of sample size planning. *Evaluation and the Health Professions, 26*(3), 258–287. doi:10.1177/0163278703255242

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9.)

See Also

`ci_c_ancova`, `ci_sc_ancova`, `ss_aipe_c`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Suppose the population error variance of some three-group ANOVA model
# is believed to be 40, and the population correlation coefficient
# of the response and the covariate is 0.22. The researcher is
# interested in the difference between the mean of group 1 and
# the average of means of group 2 and 3. To plan the sample size so
# that, with 90 percent certainty, the obtained 95 percent full
# confidence interval width is no wider than 3:

ss_aipe_c_ancova(error_var_anova = 40, rho = .22, c_weights = c(1, -0.5, -0.5),
```

```
                  width = 3, assurance = .90)
```

`ss_aipe_c_ancova_sensitivity` *Sensitivity Analysis for Sample Size Planning for the (Unstandardized) Contrast in Randomized ANCOVA From the Accuracy in Parameter Estimation (AIPE) Perspective*

Description

Performs a sensitivity analysis when planning sample size from the Accuracy in Parameter Estimation (AIPE) Perspective for the (unstandardized) contrast in randomized ANCOVA design.

Usage

```
ss_aipe_c_ancova_sensitivity(
  true_error_var_ancova = NULL,
  est_error_var_ancova = NULL,
  true_error_var_anova = NULL,
  est_error_var_anova = NULL,
  rho,
  est_rho = NULL,
  G = 10000,
  mu_y,
  sigma_y,
  mu_x,
  sigma_x,
  c_weights,
  width,
  conf_level = 0.95,
  assurance = NULL,
  filename = NULL
)
```

Arguments

`true_error_var_ancova`	population error variance of the ANCOVA model
`est_error_var_ancova`	estimated error variance of the ANCOVA model
`true_error_var_anova`	population error variance of the ANOVA model (i.e., excluding the covariate)
`est_error_var_anova`	estimated error variance of the ANOVA model (i.e., excluding the covariate)
`rho`	population correlation coefficient of the response and the covariate
`est_rho`	estimated correlation coefficient of the response and the covariate

G	number of generations (i.e., replications) of the simulation
mu_y	vector that contains the response's population mean of each group
sigma_y	the population standard deviation of the response
mu_x	the population mean of the covariate
sigma_x	the population standard deviation of the covariate
c_weights	the contrast weights
width	the desired full width of the obtained confidence interval
conf_level	the desired confidence interval coverage, (i.e., 1 - Type I error rate)
assurance	parameter to ensure that the obtained confidence interval width is narrower than the desired width with a specified degree of certainty (must be NULL or between zero and unity)
filename	an optional path for a comma separated file recording every replication (the realized contrast, its full and covariate-ignoring standard errors and their ratio, the interval width, and the three non-coverage indicators): nothing is written when filename is NULL (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at tempfile(fileext = ".csv")

Details

The arguments mu_y, mu_x, sigma_y, and sigma_x are used to generate random data in the simulations for the sensitivity analysis. The value of sigma_y should be the same as the square root of true_error_var_anova.

So far this function is based on one-covariate randomized ANCOVA design only. The argument mu_x should be a single number, because it is assumed that the population mean of the covariate is equal across groups in randomized ANCOVA.

Value

A data.frame with columns term and value summarizing the Monte Carlo sensitivity analysis across G replications. The term entries are: mean_psi, median_psi, sd_psi (summaries of the realized unstandardized contrast); mean_ci_width, median_ci_width, sd_ci_width (summaries of the realized interval widths); pct_ci_less_w (proportion of intervals narrower than the planning target width); pct_ci_miss_low and pct_ci_miss_high (tail-specific empirical non-coverage of the population contrast); total_type_I_error (overall empirical non-coverage, the sum of the two tails); mean_se_ratio (mean ratio of the contrast standard error that ignores the covariate-imbalance term to the full ANCOVA standard error); and the input echoes n_per_group, total_N, true_psi (the population contrast implied by mu_y and c_weights), est_error_var_ancova (as supplied or as resolved from est_error_var_anova and est_rho), rho, width, conf_level, and assurance (present only when an assurance was supplied). The proportion rows are on the 0 to 1 scale, not percentages. The per-replication vectors (psi_obs, se_psi, se_psi_restricted, width_obs) are not returned; they are written to the comma separated file named by filename when one is supplied.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9.)

See Also

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Monte Carlo sensitivity sweep; G is small here so the example runs quickly.
# Raise G (e.g., G = 1000 or more) for a stable sensitivity analysis.
set.seed(113)
ss_aipe_c_ancova_sensitivity(true_error_var_ancova=30,
                           est_error_var_ancova=30, rho=.2, mu_y=c(10,12,15,13), mu_x=2,
                           G=50, sigma_x=1.3, sigma_y=2, c_weights=c(1,0,-1,0), width=3)

ss_aipe_c_ancova_sensitivity(true_error_var_anova=36,
                             est_error_var_anova=36, rho=.2, est_rho=.2, G=50,
                             mu_y=c(10,12,15,13), mu_x=2, sigma_x=1.3, sigma_y=6,
                             c_weights=c(1,0,-1,0), width=3, assurance=NULL)
```

---

ss_aipe_c_sensitivity *Sensitivity Analysis for Sample Size Planning From the AIPE Perspective for an Unstandardized Contrast*

Description

Quantifies how much misspecification of the population error variance distorts an AIPE-based sample size plan for an unstandardized contrast of means. Because the half-width of the confidence interval on $\psi = \sum_j c_j \mu_j$ depends on the error variance, the contrast weights, and the per-group sample size, but not on the value of ψ itself, this sensitivity analysis varies the planning value of the error variance. On each replication the function simulates n observations per group from a normal population with variance `true_error_variance`, builds the confidence interval via `ci_c`, and summarizes the realized widths and coverage of `true_psi`.

Usage

```
ss_aipe_c_sensitivity(
  true_error_variance = NULL,
  estimated_error_variance = NULL,
  c_weights,
```

```
  width,
  true_psi = 0,
  n_per_group = NULL,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

true_error_variance	Population error variance (the data generating value). Must be positive.
estimated_error_variance	Error variance used to plan the study (the value passed to ss_aipe_c). Supply this or n_per_group but not both.
c_weights	Contrast weight vector. Must sum to zero.
width	Desired full width of the confidence interval on the unstandardized contrast.
true_psi	Population value of the contrast; the simulator places group means such that $\sum_j c_j \mu_j =$ true_psi. The width of the interval does not depend on this value but the realized coverage of true_psi does. Default 0.
n_per_group	Per-group sample size to evaluate (incompatible with estimated_error_variance); when used, the planner is bypassed.
conf_level	Confidence level (default 0.95).
assurance	Optional probability that the realized interval is no wider than width; passed to ss_aipe_c when resolving the planned sample size.
G	Number of Monte Carlo replications (default 1000).
print_iter	Logical. Print the iteration index after each replication (helpful for long runs); default FALSE.
filename	Optional path of a CSV file to receive the per-replication results (the contrast estimate, the confidence limits, the interval width, and the two tail misses), appended when the file already exists and created otherwise; the default NULL writes nothing, and a throwaway run that wants the file should point it at tempfile(fileext = ".csv").

Value

A data.frame with rows for mean / median / SD of the realized estimator and interval width, the proportion of intervals at or below width, the tail-specific and overall empirical non-coverage of true_psi, and the input echoes (per-group sample size, total sample size, true and estimated error variances, width, confidence level, and, when one was supplied, assurance).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons.)

See Also

ss_aipe_c, ci_c, ss_aipe_sc_sensitivity

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensitivity(), ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(), ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_icc_sensitivi ss_aipe_indirect_effect(), ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(), ss_aipe_omega_squared(), ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(), ss_aipe_partial_r_sensitivity(), ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(), ss_aipe_reliability_sensitivity(), ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# G = 50 keeps the example quick; a reported analysis deserves the
# default G of 1000. First the well-specified case: the planner used an
# error variance of 4, and the truth is 4.
set.seed(113)
ss_aipe_c_sensitivity(
  true_error_variance      = 4,
  estimated_error_variance = 4,
  c_weights = c(-1, 0, 1),
  width = 1, G = 50, print_iter = FALSE
)

# Misspecified: planner used 4, truth is 9. Realized widths inflate.
set.seed(113)
ss_aipe_c_sensitivity(
  true_error_variance      = 9,
  estimated_error_variance = 4,
  c_weights = c(-1, 0, 1),
  width = 1, G = 50, print_iter = FALSE
)
```

---

ss_aipe_cliff_delta *Sample Size for AIPE on Cliff's δ*

Description

Determines the sample size needed for the confidence interval on Cliff's (1993) δ (and equivalently Vargha-Delaney's $A = (\delta + 1)/2$) to have a desired width, using the maximum-variance bound on $\hat{\delta}$ (Feng & Cliff, 2004, Equation 6, p. 324).

Usage

```
ss_aipe_cliff_delta(
  delta,
  width,
  which_width = c("Full", "Lower", "Upper"),
  conf_level = 0.95,
  ratio = 1,
  assurance = NULL
)
```

Arguments

`delta`	Anticipated population Cliff's δ; numeric scalar in $(-1, 1)$.
`width`	Desired full width of the CI on δ.
`which_width`	`"Full"` (default), `"Lower"`, or `"Upper"`.
`conf_level`	Desired confidence level. Default `0.95`.
`ratio`	Ratio n_1/n_2 of the two group sample sizes. Default `1` (balanced).
`assurance`	Optional. Probability that the realized CI is no wider than `width`.

Details

Maximum-variance bound. The variance of $\hat{\delta}$ at a given δ is largest in the bimodal configuration, where it equals $(1-\delta^2)/n_b$ with n_b the bimodal group's size; for unequal groups the smaller sample size is used conservatively (Feng & Cliff, 2004, Equation 6 and following text, p. 324):

$$\mathrm{Var}(\hat{\delta}) \;\leq\; \frac{(1-\delta^2)}{\min(n_1, n_2)}.$$

Setting the half-width of a Wald-style CI $z_{1-\alpha/2}\sqrt{\mathrm{Var}(\hat{\delta})}$ equal to the target half-width and solving gives the recommended per-group sample size. The bound is conservative; the realized CI is generally narrower than the target.

Allocation. The bound is dominated by $\min(n_1, n_2)$, so balanced allocation (`ratio = 1`) is approximately optimal under standard conditions; unbalanced allocations require the larger total *N* to achieve the same precision.

Value

A `data.frame` with rows for the recommended group sample sizes n_1, n_2, the expected CI width, and the inputs echoed back.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cliff, N. (1993). Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin, 114*(3), 494–509. doi:10.1037/00332909.114.3.494

Feng, D., & Cliff, N. (2004). Monte Carlo evaluation of ordinal d with improved confidence interval. *Journal of Modern Applied Statistical Methods, 3*(2), 322–332. doi:10.22237/jmasm/1099267560

Vargha, A., & Delaney, H. D. (2000). A critique and improvement of the CL common language effect size statistics of McGraw and Wong. *Journal of Educational and Behavioral Statistics, 25*(2), 101–132. doi:10.3102/10769986025002101

See Also

cliff_delta, ss_aipe_partial_r, ss_aipe_semipartial_r

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta_sensitivity(), ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(), ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_icc_sensitivi
ss_aipe_indirect_effect(), ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(),
ss_aipe_omega_squared(), ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(), ss_aipe_partial_r_sensitivity(), ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(), ss_aipe_reliability_sensitivity(), ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# 1. Plan a balanced design so the 95% CI on delta has full width
#        <= 0.20 when anticipating delta = 0.30.
ss_aipe_cliff_delta(delta = 0.30, width = 0.20)

# 2. Unbalanced: twice as many in group 1.
ss_aipe_cliff_delta(delta = 0.30, width = 0.20, ratio = 2)
```

---

ss_aipe_cliff_delta_sensitivity

Sensitivity Analysis for Sample Size Planning From the AIPE Perspective for Cliff's Delta

---

Description

Quantifies how much misspecification of the population Cliff's delta ($\delta = \Pr(X > Y) - \Pr(X < Y)$) distorts an AIPE-based sample size plan. On each replication the function simulates two independent samples whose population Cliff's delta equals `true_delta`, computes the sample `cliff_delta` and its CI, and summarizes the realized widths and coverage.

Data generating mechanism. The simulator draws each sample from a normal distribution and chooses the mean shift so that the implied Cliff's delta equals `true_delta`. For normal samples $\delta = 2\Phi(\Delta/\sqrt{2}) - 1$ where Δ is the standardized mean difference, so the simulator sets $\Delta = \sqrt{2} \cdot \Phi^{-1}((1+\delta)/2)$.

Usage

```
ss_aipe_cliff_delta_sensitivity(
  true_delta = NULL,
  estimated_delta = NULL,
  ratio = 1,
  width,
  specified_N = NULL,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

`true_delta`	Population Cliff's delta (the data generating value); in $(-1, 1)$.
`estimated_delta`	Planning value passed to `ss_aipe_cliff_delta`; supply this or `specified_N` but not both.
`ratio`	Allocation ratio n_1/n_2 (default 1).
`width`	Desired full width of the CI on Cliff's delta.
`specified_N`	Total sample size to evaluate (split per `ratio`).
`conf_level`	Confidence level (default `0.95`).
`assurance`	Optional assurance probability.
`G`	Number of Monte Carlo replications.
`print_iter`	Logical.
`filename`	Optional path for a comma separated file recording every replication (the sample Cliff's delta, the two confidence limits, the interval width, and two indicators of whether the interval missed `true_delta` below or above): nothing is written when `filename` is `NULL` (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at `tempfile(fileext = ".csv")`.

Value

A `data.frame` with rows for mean / median / SD of the realized Cliff's delta and CI width, the proportion of intervals at or below `width`, tail-specific and overall non-coverage of `true_delta`, and the input echoes, including `assurance` (present only when an assurance was supplied).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cliff, N. (1993). Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin, 114*(3), 494–509. doi:10.1037/00332909.114.3.494

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

`ss_aipe_cliff_delta`, `cliff_delta`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: `ss_aipe_c_sensitivity()`, `ss_aipe_cliff_delta()`, `ss_aipe_composite_sem()`, `ss_aipe_equivalence_r()`, `ss_aipe_equivalence_r_sensitivity()`, `ss_aipe_equivalence_smd()`, `ss_aipe_equivalence_smd_sensitivity()`, `ss_aipe_icc()`, `ss_aipe_icc_sensitivity()`, `ss_aipe_indirect_effec`
`ss_aipe_indirect_effect_sensitivity()`, `ss_aipe_mixed_effects_sensitivity()`, `ss_aipe_omega_squared()`, `ss_aipe_omega_squared_sensitivity()`, `ss_aipe_partial_r()`, `ss_aipe_partial_r_sensitivity()`, `ss_aipe_pcm_sensitivity()`, `ss_aipe_r()`, `ss_aipe_r_sensitivity()`, `ss_aipe_reliability_sensitivity()`, `ss_aipe_semipartial_r()`, `ss_aipe_semipartial_r_sensitivity()`

Examples

```
set.seed(113)
# Small G keeps the Monte Carlo sweep fast; raise G for a real plan.
ss_aipe_cliff_delta_sensitivity(
  true_delta = 0.30, estimated_delta = 0.30,
  width = 0.30, G = 25, print_iter = FALSE
)
```

`ss_aipe_composite_sem` *Sample Size for Accurate Estimation of a Set of SEM Parameters*

Description

Determine the necessary sample size for a structural equation model study so that the confidence interval for every parameter of interest is sufficiently narrow in the same study, or, given a sample size, return how narrow the set of intervals can be expected to be. This is the accuracy in parameter estimation (AIPE) counterpart of `ss_power_composite_sem`: where that function plans for every parameter to be statistically significant jointly, this one plans for every parameter to be estimated with a confidence interval no wider than desired, the goal when the research questions concern the magnitudes of the effects rather than their existence. The parameters of interest are any labeled parameters of a lavaan analysis model, structural paths, loadings, covariances, or quantities defined with `:=` such as an indirect effect, and any subset of them can make up the set.

Usage

```
ss_aipe_composite_sem(
  model,
  Sigma = NULL,
  pop_model = NULL,
  mu = NULL,
  parameters = NULL,
  desired_width,
  conf_level = 0.95,
  assurance = NULL,
  N = NULL,
  G = 1000,
  seed = NULL,
  ...
)
```

Arguments

`model`	A single character string giving the free analysis model in lavaan model syntax (see `model.syntax`), the model that would be fit to the data. Each parameter of interest must carry a parameter label so it can be referred to by name, for example `"f2 ~ b*f1"` labels the structural path `b`, and `"ab := a*b"` defines an indirect effect from the labeled paths `a` and `b`.
`Sigma`	Population covariance matrix of the observed variables, with row and column names matching the observed variables in `model`. It is typically obtained from a fully fixed population model via `cov_sem`. Supply exactly one of `Sigma` or `pop_model`.
`pop_model`	A single character string giving the population model in lavaan model syntax with every parameter fixed to its population value, from which `cov_sem` derives `Sigma` (and the population means, when the model has a mean structure). The fixed values are values the researcher posits (from theory, prior studies, or pilot data), never sample estimates. Supply exactly one of `Sigma` or `pop_model`.
`mu`	Optional population means of the observed variables, used with `Sigma`: a named numeric vector with one entry per observed variable, or an unnamed vector in the row order of `Sigma`. The default `NULL` is zero means. Means matter only

when the analysis model has a mean structure (an intercept term such as `s ~ 1`, as in a latent growth curve model); when `pop_model` is supplied its mean structure provides the means and `mu` must not also be given.

`parameters` Character vector of the parameter labels that make up the set. The default `NULL` uses every user-labeled parameter in `model`, in order of appearance, so labeling exactly the parameters of interest is the simplest way to state the set.

`desired_width` The desired full confidence interval width for each parameter of interest: a single value applied to every parameter, or a named numeric vector with one entry per parameter label. An unnamed vector of several widths is not accepted, so a width can never silently attach to the wrong parameter.

`conf_level` Confidence level of each interval (default 0.95).

`assurance` The desired probability that a single study yields confidence intervals no wider than desired for every parameter of interest simultaneously (a value in [0.5, 1)), or `NULL` (the default) to plan against the expected widths instead; see Details.

`N` Sample size; if supplied, the realized interval widths at that *N* are summarized rather than a sample size planned.

`G` Number of converged Monte Carlo replications per evaluated sample size (default 1000). The simulation error of each estimated proportion is about $\sqrt{p(1-p)/G}$; raise `G` for a sharper answer.

`seed` Optional integer seed for reproducibility. The default `NULL` uses the current state of the random number generator; a supplied seed is set internally and the prior state restored on exit.

`...` Additional arguments passed to `sem`, both when the population values are resolved and for every Monte Carlo fit (for example `std.lv = TRUE` or `missing = "listwise"`). A robust estimator such as `"MLM"` cannot be used here: the same arguments reach the setup fit, which is always from summary statistics, and lavaan refuses a robust estimator there. The Monte Carlo data are multivariate normal by construction, so a robust estimator would buy nothing.

Details

AIPE planning for a single targeted SEM parameter is available in closed form (Lai & Kelley, 2011; `ss_aipe_sem_path`), but most studies estimate several effects and report all of them; a design is only as informative as its widest interval of interest. This function plans for the set by a priori Monte Carlo simulation (Muthén & Muthén, 2002; Maxwell, Kelley, & Rausch, 2008): for a candidate *N*, `G` data sets are drawn from the multivariate normal population with covariance matrix `Sigma`, the analysis model is fit to each, and each parameter's Wald confidence interval width, twice $z_{1-\alpha/2}$ times its standard error, is recorded. Because the estimates share one fitted model, the widths are dependent; the simulation reflects that dependence exactly, at the stated *N*, with no asymptotic shortcut.

Two planning criteria are available. With `assurance = NULL` the necessary sample size is the smallest *N* at which the mean simulated width of every parameter's interval is at or below its desired width, the expected-width criterion of the AIPE framework applied to each member of the set. Widths vary from sample to sample around their means, so each interval separately lands at or below its desired width in roughly half of the realizations. That is a statement about one interval at a time, not about the set: the probability that *every* interval is narrow enough at once falls well below

one half as soon as more than one parameter binds, and falls further the more parameters are targeted and the more weakly their widths move together. Planning the whole set to a stated probability is exactly what `assurance` is for. Supplying `assurance` plans against the joint event instead: the smallest *N* at which the proportion of replications where every interval is simultaneously within its desired width reaches the assurance. The joint event is contained in each marginal event, so its probability is at most the smallest per-parameter proportion, and the `width_within_desired_<label>` rows show which parameter binds the design.

When `N` is `NULL` the search starts at the largest of the per-parameter closed-form sample sizes (the no-assurance approximation `ss_aipe_sem_path` uses, computed from the asymptotic variances before any simulation), brackets the crossing geometrically, and bisects to adjacent integers, each candidate evaluated with its own `G` replications. A planning call therefore fits the analysis model several thousand times at the default `G`, and even at the smallest admissible `G` the search runs for several seconds, so the example below evaluates a stated `N`, which is the cheap half of the method. A planning call is the same call with `N` left out: supplying `assurance` plans against the joint event, and leaving `assurance` out as well plans against the expected widths; either way the first row of the result is `necessary_N` rather than `specified_N`. The vignette `vignette("composite_sem_planning", package = "DMAR")` works through both planning calls for a mediation model and a latent growth curve model, with reference values computed at `G = 10000`.

Each reported proportion carries a simulation standard error of about $\sqrt{p(1-p)/G}$, and the necessary sample size inherits that uncertainty; raising `G` narrows it, and reporting the seed makes a plan reproducible.

Value

A `data.frame` (a `dmar_tbl`) with `term` and `value` columns: the `necessary_N` (or supplied `specified_N`), the `composite_assurance` (the proportion of replications in which every interval was simultaneously within its desired width, reported under both criteria), then for each parameter its `mean_width_<label>`, its marginal `width_within_desired_<label>` proportion, its `desired_width_<label>`, and its purported `population_<label>` value under the analysis model, followed by `conf_level`, the requested `replications`, the `converged_replications` the summary is based on, and, when supplied, the `assurance`.

Note

A replication whose fit does not converge, or converges without a usable standard error for some parameter of interest, is discarded and fresh data are drawn, up to `20 * G` attempts per evaluated sample size; the reported summaries condition on convergence. When fewer than `G` replications converge within the cap, a single warning is issued and the summary is based on the converged replications (their count is the `converged_replications` row).

Because the planner itself is a Monte Carlo study, it has no separate `_sensitivity` sibling; to study misspecification of the population values, rerun the planner with the alternative `Sigma` or `pop_model` values under consideration and compare the plans.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Lai, K., & Kelley, K. (2011). Accuracy in parameter estimation for targeted effects in structural equation modeling: Sample size planning for narrow confidence intervals. *Psychological Methods, 16*(2), 127–148. doi:10.1037/a0021764

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

Muthén, L. K., & Muthén, B. O. (2002). How to use a Monte Carlo study to decide on sample size and determine power. *Structural Equation Modeling, 9*(4), 599–620. doi:10.1207/S15328007SEM0904_8

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48*(2), 1–36. doi:10.18637/jss.v048.i02

See Also

`ss_power_composite_sem` for the same set of parameters planned for joint statistical significance; `cov_sem` for deriving `Sigma` from a fully fixed population model; `ss_aipe_sem_path` and `ss_aipe_sem_path_sensitivity` for a single targeted path.

Other AIPE sample size planning: `ss_aipe_c_sensitivity()`, `ss_aipe_cliff_delta()`, `ss_aipe_cliff_delta_sensit`
`ss_aipe_equivalence_r()`, `ss_aipe_equivalence_r_sensitivity()`, `ss_aipe_equivalence_smd()`,
`ss_aipe_equivalence_smd_sensitivity()`, `ss_aipe_icc()`, `ss_aipe_icc_sensitivity()`, `ss_aipe_indirect_effec`
`ss_aipe_indirect_effect_sensitivity()`, `ss_aipe_mixed_effects_sensitivity()`, `ss_aipe_omega_squared()`,
`ss_aipe_omega_squared_sensitivity()`, `ss_aipe_partial_r()`, `ss_aipe_partial_r_sensitivity()`,
`ss_aipe_pcm_sensitivity()`, `ss_aipe_r()`, `ss_aipe_r_sensitivity()`, `ss_aipe_reliability_sensitivity()`,
`ss_aipe_semipartial_r()`, `ss_aipe_semipartial_r_sensitivity()`

Examples

```
# A mediation model whose research questions concern the magnitudes of
# both individual paths and the indirect effect. The population model
# fixes every parameter to its purported population value.
pop_model <- "
  f1 =~ 1*y1 + 0.8*y2 + 0.8*y3
  f2 =~ 1*y4 + 0.8*y5 + 0.8*y6
  f3 =~ 1*y7 + 0.8*y8 + 0.8*y9
  f2 ~ 0.4*f1
  f3 ~ 0.5*f2 + 0.2*f1
  f1 ~~ 1*f1
  f2 ~~ 0.84*f2
  f3 ~~ 0.7*f3
  y1 ~~ 0.5*y1; y2 ~~ 0.5*y2; y3 ~~ 0.5*y3
  y4 ~~ 0.5*y4; y5 ~~ 0.5*y5; y6 ~~ 0.5*y6
  y7 ~~ 0.5*y7; y8 ~~ 0.5*y8; y9 ~~ 0.5*y9
"


# The analysis model labels the two paths and defines the indirect
# effect; all three make up the set of interest.
```

```
analysis_model <- "
  f1 =~ y1 + y2 + y3
  f2 =~ y4 + y5 + y6
  f3 =~ y7 + y8 + y9
  f2 ~ a*f1
  f3 ~ b*f2 + cp*f1
  ab := a*b
"

# Realized interval widths at N = 200, with the indirect effect held to a
# narrower interval than the paths through a named vector of widths. Each
# interval lands within its desired width in most of the replications, yet
# all three do so together in far fewer of them: that joint proportion,
# reported as composite_assurance, is what a design of this kind has to be
# planned against. G = 20 keeps the example quick; a reported plan
# deserves the default G = 1000 or more.
ss_aipe_composite_sem(model = analysis_model, pop_model = pop_model,
                      parameters = c("a", "b", "ab"),
                      desired_width = c(a = 0.35, b = 0.40, ab = 0.25),
                      N = 200, G = 20, seed = 113)
```

`ss_aipe_crd` *Find Target Sample Sizes for the Accuracy in Unstandardized Conditions Means Estimation in CRD*

Description

Find target sample sizes (the number of clusters, cluster size, or both) for the accuracy in unstandardized conditions means estimation in CRD. If users wish to seek for both types of sample sizes simultaneously, an additional constraint is required, such as a desired width or a desired budget.

Usage

```
ss_aipe_crd_n_clusters_fixed_width(
  width,
  n_individuals,
  pr_treat,
  tau_Y = NULL,
  sigma2_Y = NULL,
  total_var = NULL,
  icc_Y = NULL,
  R2_between = 0,
  R2_within = 0,
  num_predictors = 0,
  assurance = NULL,
  conf_level = 0.95,
  clus_cost = NULL,
  indiv_cost = NULL,
```

```
  diff_size = NULL
)

ss_aipe_crd_n_individuals_fixed_width(
  width,
  n_clusters,
  pr_treat,
  tau_Y = NULL,
  sigma2_Y = NULL,
  total_var = NULL,
  icc_Y = NULL,
  R2_between = 0,
  R2_within = 0,
  num_predictors = 0,
  assurance = NULL,
  conf_level = 0.95,
  clus_cost = NULL,
  indiv_cost = NULL,
  diff_size = NULL
)

ss_aipe_crd_n_clusters_fixed_budget(
  budget,
  n_individuals,
  clus_cost = 0,
  indiv_cost = 1,
  pr_treat = NULL,
  tau_Y = NULL,
  sigma2_Y = NULL,
  total_var = NULL,
  icc_Y = NULL,
  R2_between = 0,
  R2_within = 0,
  num_predictors = 0,
  assurance = NULL,
  conf_level = 0.95,
  diff_size = NULL
)

ss_aipe_crd_n_individuals_fixed_budget(
  budget,
  n_clusters,
  clus_cost = 0,
  indiv_cost = 1,
  pr_treat = NULL,
  tau_Y = NULL,
  sigma2_Y = NULL,
  total_var = NULL,
```

```
  icc_Y = NULL,
  R2_between = 0,
  R2_within = 0,
  num_predictors = 0,
  assurance = NULL,
  conf_level = 0.95,
  diff_size = NULL
)

ss_aipe_crd_both_fixed_budget(
  budget,
  clus_cost = 0,
  indiv_cost = 1,
  pr_treat,
  tau_Y = NULL,
  sigma2_Y = NULL,
  total_var = NULL,
  icc_Y = NULL,
  R2_between = 0,
  R2_within = 0,
  num_predictors = 0,
  assurance = NULL,
  conf_level = 0.95,
  diff_size = NULL
)

ss_aipe_crd_both_fixed_width(
  width,
  clus_cost = 0,
  indiv_cost = 1,
  pr_treat,
  tau_Y = NULL,
  sigma2_Y = NULL,
  total_var = NULL,
  icc_Y = NULL,
  R2_between = 0,
  R2_within = 0,
  num_predictors = 0,
  assurance = NULL,
  conf_level = 0.95,
  diff_size = NULL
)
```

Arguments

width	The desired width of the confidence interval of the unstandardized means difference
n_individuals	The number of individuals in each cluster (cluster size)

pr_treat	The proportion of treatment clusters
tau_Y	The residual variance in the between level before accounting for the covariate
sigma2_Y	The residual variance in the within level before accounting for the covariate
total_var	The total residual variance before accounting for the covariate
icc_Y	The intraclass correlation of the dependent variable
R2_between	The proportion of variance explained in the between level (used when covariate = TRUE)
R2_within	The proportion of variance explained in the within level (used when covariate = TRUE)
num_predictors	The number of predictors used in the between level
assurance	The degree of assurance, which is the value with which confidence can be placed that describes the likelihood of obtaining a confidence interval less than the value specified (e.g., .80, .90, .95)
conf_level	The desired level of confidence for the confidence interval
clus_cost	The cost of collecting a new cluster regardless of the number of individuals collected in each cluster
indiv_cost	The cost of collecting a new individual
diff_size	Difference cluster size specification. The differences in cluster sizes can be specified in two ways, and the specified vector is recycled across the clusters. First, users may specify differences as integers, which can be negative or positive; the resulting cluster sizes add the specified values to the estimated cluster size. For example, if the cluster size is 25, the number of clusters is 10, and diff_size = c(-1, 0, 1), the cluster sizes will be 24, 25, 26, 24, 25, 26, 24, 25, 26, and 24. Second, users may specify multipliers of the cluster size as positive decimals; at least one value must be non-integer, which is what selects the multiplicative form. The resulting cluster sizes multiply the estimated cluster size by the specified values and round to the nearest integer. For example, if the cluster size is 25, the number of clusters is 10, and diff_size = c(0.8, 1, 1.2), the cluster sizes will be 20, 25, 30, 20, 25, 30, 20, 25, 30, and 20. In either form a resulting cluster size below 1 is set to 1. If NULL, the cluster size is equal across clusters
n_clusters	The desired number of clusters
budget	The desired amount of budget

Details

Here are the functions' descriptions:

ss_aipe_crd_n_clusters_fixed_width Find the number of clusters given a specified width of the confidence interval and the cluster size

ss_aipe_crd_n_individuals_fixed_width Find the cluster size given a specified width of the confidence interval and the number of clusters

ss_aipe_crd_n_clusters_fixed_budget Find the number of clusters given a budget and the cluster size

`ss_aipe_crd_n_individuals_fixed_budget` Find the cluster size given a budget and the number of clusters

`ss_aipe_crd_both_fixed_budget` Find the sample size combinations (the number of clusters and that cluster size) providing the narrowest confidence interval given the fixed budget

`ss_aipe_crd_both_fixed_width` Find the sample size combinations (the number of clusters and that cluster size) providing the lowest cost given the specified width of the confidence interval

Value

The `ss_aipe_crd_n_clusters_fixed_width` and `ss_aipe_crd_n_clusters_fixed_budget` functions provide the number of clusters. The `ss_aipe_crd_n_individuals_fixed_width` and `ss_aipe_crd_n_individuals_` functions provide the cluster size. The `ss_aipe_crd_both_fixed_budget` and `ss_aipe_crd_both_fixed_width` provide the number of clusters and the cluster size, respectively.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Pornprasertmanit, S., & Schneider, W. J. (2014). Accuracy in parameter estimation in cluster randomized designs. *Psychological Methods, 19*(3), 356–379. doi:10.1037/a0037036

See Also

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Examples for each function
ss_aipe_crd_n_clusters_fixed_width(width = 0.3, n_individuals = 30,
  pr_treat = 0.5, tau_Y = 0.25, sigma2_Y = 0.75)

ss_aipe_crd_n_individuals_fixed_width(width = 0.3, n_clusters = 250,
  pr_treat = 0.5, tau_Y = 0.25, sigma2_Y = 0.75)

ss_aipe_crd_n_clusters_fixed_budget(budget = 10000, n_individuals = 20,
  clus_cost = 20, indiv_cost = 1)

ss_aipe_crd_n_individuals_fixed_budget(budget = 10000, n_clusters = 30,
  clus_cost = 20, indiv_cost = 1,
  pr_treat = 0.5, tau_Y = 0.05, sigma2_Y = 0.95, assurance = 0.8)

ss_aipe_crd_both_fixed_budget(budget = 10000, clus_cost = 30, indiv_cost = 1,
  pr_treat = 0.5, tau_Y = 0.25, sigma2_Y = 0.75)

ss_aipe_crd_both_fixed_width(width = 0.3, clus_cost = 0, indiv_cost = 1,
  pr_treat = 0.5, tau_Y = 0.25, sigma2_Y = 0.75)

# Examples for different cluster size
```

```
set.seed(113)
ss_aipe_crd_n_clusters_fixed_width(width = 0.3, n_individuals = 30,
  pr_treat = 0.5, tau_Y = 0.25, sigma2_Y = 0.75,
  diff_size = c(-2, 1, 0, 2, -1, 3, -3, 0, 0))

# Examples for different number of clusters
ss_aipe_crd_n_individuals_fixed_width(width = 0.3, n_clusters = 250,
  pr_treat = 0.5, tau_Y = 0.25, sigma2_Y = 0.75,
  diff_size = c(0.6, 1.2, 0.8, 1.4, 1, 1, 1.1, 0.9))
```

`ss_aipe_crd_es` *Find Target Sample Sizes for the Accuracy in Standardized Conditions Means Estimation in CRD*

Description

Find target sample sizes (the number of clusters, cluster size, or both) for the accuracy in standardized conditions means estimation in CRD. If users wish to seek for both types of sample sizes simultaneously, an additional constraint is required, such as a desired width or a desired budget. This function uses the likelihood-based confidence interval (Cheung, 2009) by the `OpenMx` package (Boker et al., 2011). See further details at Pornprasertmanit and Schneider (2014).

Usage

```
ss_aipe_crd_es_n_clusters_fixed_width(
  width,
  n_individuals,
  es,
  es_type = 1,
  icc_Y,
  pr_treat,
  R2_between = 0,
  R2_within = 0,
  num_predictors = 0,
  assurance = NULL,
  conf_level = 0.95,
  nrep = 1000,
  icc_Z = NULL,
  seed = NULL,
  multicore = FALSE,
  num_proc = NULL,
  clus_cost = NULL,
  indiv_cost = NULL,
  diff_size = NULL
)

ss_aipe_crd_es_n_individuals_fixed_width(
```

```
  width,
  n_clusters,
  es,
  es_type = 1,
  icc_Y,
  pr_treat,
  R2_between = 0,
  R2_within = 0,
  num_predictors = 0,
  assurance = NULL,
  conf_level = 0.95,
  nrep = 1000,
  icc_Z = NULL,
  seed = NULL,
  multicore = FALSE,
  num_proc = NULL,
  clus_cost = NULL,
  indiv_cost = NULL,
  diff_size = NULL
)

ss_aipe_crd_es_n_clusters_fixed_budget(
  budget,
  n_individuals,
  clus_cost,
  indiv_cost,
  nrep = NULL,
  pr_treat = NULL,
  icc_Y = NULL,
  es = NULL,
  es_type = 1,
  num_predictors = 0,
  icc_Z = NULL,
  R2_within = NULL,
  R2_between = NULL,
  assurance = NULL,
  seed = NULL,
  multicore = FALSE,
  num_proc = NULL,
  conf_level = 0.95,
  diff_size = NULL
)

ss_aipe_crd_es_n_individuals_fixed_budget(
  budget,
  n_clusters,
  clus_cost,
  indiv_cost,
```

```
  nrep = NULL,
  pr_treat = NULL,
  icc_Y = NULL,
  es = NULL,
  es_type = 1,
  num_predictors = 0,
  icc_Z = NULL,
  R2_within = NULL,
  R2_between = NULL,
  assurance = NULL,
  seed = NULL,
  multicore = FALSE,
  num_proc = NULL,
  conf_level = 0.95,
  diff_size = NULL
)

ss_aipe_crd_es_both_fixed_budget(
  budget,
  clus_cost = 0,
  indiv_cost = 1,
  es,
  es_type = 1,
  icc_Y,
  pr_treat,
  R2_between = 0,
  R2_within = 0,
  num_predictors = 0,
  assurance = NULL,
  conf_level = 0.95,
  nrep = 1000,
  icc_Z = NULL,
  seed = NULL,
  multicore = FALSE,
  num_proc = NULL,
  diff_size = NULL
)

ss_aipe_crd_es_both_fixed_width(
  width,
  clus_cost = 0,
  indiv_cost = 1,
  es,
  es_type = 1,
  icc_Y,
  pr_treat,
  R2_between = 0,
  R2_within = 0,
```

```
  num_predictors = 0,
  assurance = NULL,
  conf_level = 0.95,
  nrep = 1000,
  icc_Z = NULL,
  seed = NULL,
  multicore = FALSE,
  num_proc = NULL,
  diff_size = NULL
)
```

Arguments

width	The desired width of the confidence interval of the unstandardized means difference
n_individuals	The number of individuals in each cluster (cluster size)
es	The amount of effect size
es_type	The type of effect size. There are only three possible options: 0 = the effect size using total standard deviation, 1 = the effect size using the individual-level standard deviation (level 1), 2 = the effect size using the cluster-level standard deviation (level 2)
icc_Y	The intraclass correlation of the dependent variable
pr_treat	The proportion of treatment clusters
R2_between	The proportion of variance explained in the between level (used when covariate = TRUE)
R2_within	The proportion of variance explained in the within level (used when covariate = TRUE)
num_predictors	The number of predictors used in the between level
assurance	The degree of assurance, which is the value with which confidence can be placed that describes the likelihood of obtaining a confidence interval less than the value specified (e.g., .80, .90, .95)
conf_level	The desired level of confidence for the confidence interval
nrep	The number of replications used in a priori Monte Carlo simulation
icc_Z	The intraclass correlation of the covariate (used when covariate = TRUE). If icc_Z = 0, the within-level covariate will be only used. If icc_Z = 1, the between-level covariate will be only used
seed	An optional integer seed for the a priori Monte Carlo simulation. The default NULL uses the current state of the random number generator and leaves it unchanged, so repeated calls reflect the genuine sampling variability of the simulation. Supply an integer for reproducible results, in which case the generator state is restored on exit
multicore	Use multiple processors within a computer. Specify as TRUE to use it
num_proc	The number of processors to be used when multicore = TRUE. If it is not specified, the package will use the maximum number of processors in a machine

clus_cost	The cost of collecting a new cluster regardless of the number of individuals collected in each cluster
indiv_cost	The cost of collecting a new individual
diff_size	Difference cluster size specification. The differences in cluster sizes can be specified in two ways, and the specified vector is recycled across the clusters. First, users may specify differences as integers, which can be negative or positive; the resulting cluster sizes add the specified values to the estimated cluster size. For example, if the cluster size is 25, the number of clusters is 10, and `diff_size = c(-1, 0, 1)`, the cluster sizes will be 24, 25, 26, 24, 25, 26, 24, 25, 26, and 24. Second, users may specify multipliers of the cluster size as positive decimals; at least one value must be non-integer, which is what selects the multiplicative form. The resulting cluster sizes multiply the estimated cluster size by the specified values and round to the nearest integer. For example, if the cluster size is 25, the number of clusters is 10, and `diff_size = c(0.8, 1, 1.2)`, the cluster sizes will be 20, 25, 30, 20, 25, 30, 20, 25, 30, and 20. In either form a resulting cluster size below 1 is set to 1. If `NULL`, the cluster size is equal across clusters
n_clusters	The desired number of clusters
budget	The desired amount of budget

Details

Here are the functions' descriptions:

`ss_aipe_crd_es_n_clusters_fixed_width` Find the number of clusters given a specified width of the confidence interval and the cluster size

`ss_aipe_crd_es_n_individuals_fixed_width` Find the cluster size given a specified width of the confidence interval and the number of clusters

`ss_aipe_crd_es_n_clusters_fixed_budget` Find the number of clusters given a budget and the cluster size

`ss_aipe_crd_es_n_individuals_fixed_budget` Find the cluster size given a budget and the number of clusters

`ss_aipe_crd_es_both_fixed_budget` Find the sample size combinations (the number of clusters and that cluster size) providing the narrowest confidence interval given the fixed budget

`ss_aipe_crd_es_both_fixed_width` Find the sample size combinations (the number of clusters and that cluster size) providing the lowest cost given the specified width of the confidence interval

Every answer that targets a width, and the expected width a budget planner reports when `nrep` and the population values are supplied, rests on an a priori Monte Carlo simulation: a candidate design is evaluated by generating `nrep` data sets and reading the likelihood-based confidence interval on the standardized effect size from OpenMx. The planners whose answer needs one such evaluation, or a handful, run in about a second at a small `nrep` and are the ones shown in the examples. The two that search over the number of clusters, `ss_aipe_crd_es_n_clusters_fixed_width` at a fixed cluster size and `ss_aipe_crd_es_both_fixed_width`, which repeats that search across candidate cluster sizes and keeps the least costly combination that reaches the width, evaluate many candidates in turn and run for minutes at the default `nrep = 1000`, so they are not among the examples. A call such as `ss_aipe_crd_es_n_clusters_fixed_width(width = 0.3, n_individuals`

= 20, es = 0.5,es_type = 1, icc_Y = 0.25, pr_treat = 0.5, nrep = 1000, seed = 113) returns the number of clusters of 20 individuals that brings the expected width of the interval on the individual-level standardized effect size to 0.3, and ss_aipe_crd_es_both_fixed_width(width = 0.5, clus_cost = 5, indiv_cost = 1, es = 0.5,es_type = 1, icc_Y = 0.25, pr_treat = 0.5, nrep = 1000, seed = 113) returns the least costly pairing of clusters and cluster size at those costs that reaches a width of 0.5. Both accept diff_size for unequal cluster sizes, in which case the simulated data sets carry the unequal sizes and the confidence interval comes from a multiple-group model.

Value

The ss_aipe_crd_es_n_clusters_fixed_width and ss_aipe_crd_es_n_clusters_fixed_budget functions provide the number of clusters. The ss_aipe_crd_es_n_individuals_fixed_width and ss_aipe_crd_es_n_individuals_fixed_budget functions provide the cluster size. The ss_aipe_crd_es_both_fixe and ss_aipe_crd_es_both_fixed_width provide the number of clusters and the cluster size, respectively.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Boker, S. M., Neale, M. C., Maes, H. H., Wilde, M., Spiegel, M., Brick, T. R., ... Fox, J. (2011). OpenMx: An open source extended structural equation modeling framework. *Psychometrika, 76*(2), 306–317. doi:10.1007/s11336-010-9200-6

Cheung, M. W.-L. (2009). Constructing approximate confidence intervals for parameters with structural equation models. *Structural Equation Modeling, 16*(2), 267–294. doi:10.1080/10705510902751291

Pornprasertmanit, S., & Schneider, W. J. (2010). *Efficient sample size for power and desired accuracy in Cohen's d estimation in two-group cluster randomized design* (Master Thesis). Illinois State University, Normal, IL.

Pornprasertmanit, S., & Schneider, W. J. (2014). Accuracy in parameter estimation in cluster randomized designs. *Psychological Methods, 19*(3), 356–379. doi:10.1037/a0037036

See Also

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Two of these planners answer a question the budget alone settles. Given
# what it costs to open a cluster and what it costs to collect one more
# individual, the first reports how many clusters a budget buys at a fixed
# cluster size and the second reports how large each cluster can be at a
# fixed number of clusters. Clusters cost nothing to open here and each
# individual costs 1, so the budget buys 1000 individuals and the only
# question is how to arrange them.
ss_aipe_crd_es_n_clusters_fixed_budget(budget = 1000, n_individuals = 20,
```

```
  clus_cost = 0, indiv_cost = 1)

ss_aipe_crd_es_n_individuals_fixed_budget(budget = 1000, n_clusters = 200,
  clus_cost = 0, indiv_cost = 1)

# The interval width these planners can report, and every answer that
# targets a width, rests on an a priori Monte Carlo simulation: a candidate
# design is evaluated by generating nrep data sets and reading the
# likelihood-based confidence interval on the standardized effect size from
# OpenMx. The calls below use nrep = 2 so that the page runs quickly; a
# reported plan deserves the default nrep = 1000. Each call describes a
# population standardized effect size of 0.5, with es_type = 1 putting that
# effect size in individual-level standard deviation units and a quarter of
# the outcome variance lying between clusters.

# Supplying nrep and the population values to a budget planner adds the
# expected width of the interval the affordable design buys.
ss_aipe_crd_es_n_clusters_fixed_budget(budget = 1000, n_individuals = 20,
  clus_cost = 0, indiv_cost = 1, es = 0.5, es_type = 1, icc_Y = 0.25,
  pr_treat = 0.5, nrep = 2, seed = 113)

# Cluster size needed for a target width, given the number of clusters.
# With 250 clusters the planner settles on the smallest cluster size it
# will consider, two individuals per cluster, and the expected width still
# comes in well under the target: for a contrast between conditions that
# are assigned at the cluster level, precision is bought with clusters
# rather than with what happens inside them.
ss_aipe_crd_es_n_individuals_fixed_width(width = 0.5, n_clusters = 250,
  es = 0.5, es_type = 1, icc_Y = 0.25, pr_treat = 0.5, nrep = 2,
  seed = 113)

# Once recruiting a cluster costs 5, the number of clusters and the cluster
# size trade off against each other, and this planner searches the
# combinations the budget allows for the narrowest expected interval.
ss_aipe_crd_es_both_fixed_budget(budget = 1000, clus_cost = 5,
  indiv_cost = 1, es = 0.5, es_type = 1, icc_Y = 0.25, pr_treat = 0.5,
  nrep = 2, seed = 113)

# Unequal cluster sizes. Every planner accepts diff_size, which gives each
# cluster's deviation from n_individuals as an integer offset or as a
# multiplicative factor, recycled across the clusters; the planner prints
# the resulting cluster sizes and their frequencies ahead of its table.
# The budget planner shows both forms here.
ss_aipe_crd_es_n_clusters_fixed_budget(budget = 1000, n_individuals = 20,
  clus_cost = 0, indiv_cost = 1, diff_size = c(-2, 1, 0, 2, -1, 3, -3, 0, 0))

ss_aipe_crd_es_n_clusters_fixed_budget(budget = 1000, n_individuals = 20,
  clus_cost = 0, indiv_cost = 1,
  diff_size = c(0.6, 1.2, 0.8, 1.4, 1, 1, 1.1, 0.9))
```

ss_aipe_cv *Sample Size Planning for the Coefficient of Variation Given the Goal of Accuracy in Parameter Estimation Approach to Sample Size Planning*

Description

Determines the necessary sample size so that the expected confidence interval width for the coefficient of variation will be sufficiently narrow, optionally with a desired degree of certainty that the interval will not be wider than desired. The population coefficient of variation may be given directly as `C_of_V` or through `mu` and `sigma`, in which case `C_of_V` is taken as `sigma / mu`. The value of `C_of_V` should be positive.

Usage

```
ss_aipe_cv(
  C_of_V = NULL,
  width = NULL,
  conf_level = 0.95,
  assurance = NULL,
  mu = NULL,
  sigma = NULL,
  alpha_lower = NULL,
  alpha_upper = NULL,
  ...
)
```

Arguments

`C_of_V`	Population coefficient of variation on which the sample size procedure is based
`width`	Desired (full) width of the confidence interval
`conf_level`	Confidence interval coverage; 1-Type I error rate
`assurance`	Value with which confidence can be placed that describes the likelihood of obtaining a confidence interval less than the value specified (e.g., .80, .90, .95)
`mu`	Population mean (specified with `sigma` when `C_of_V` is not specified)
`sigma`	Population standard deviation (specified with `mu` when `C_of_V` is not specified)
`alpha_lower`	Type I error for the lower confidence limit
`alpha_upper`	Type I error for the upper confidence limit
`...`	For modifying parameters of functions this function calls

Value

Returns the necessary sample size given the input specifications.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Chattopadhyay, B., & Kelley, K. (2016). Estimation of the coefficient of variation with minimum risk: A sequential method for minimizing sampling error and study cost. *Multivariate Behavioral Research, 51*(5), 627–648. doi:10.1080/00273171.2016.1203279

Kelley, K. (2007). Sample size planning for the coefficient of variation from the accuracy in parameter estimation approach. *Behavior Research Methods, 39*(4), 755–766. doi:10.3758/BF03192966

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3.)

See Also

`ss_aipe_cv_sensitivity`, `cv`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Suppose one wishes to have a confidence interval with an expected width of .10
# for a 99% confidence interval when the population coefficient of variation is .10.
ss_aipe_cv(C_of_V = .1, width = .1, conf_level = .99)

# The same planning problem parameterized by the population mean and standard
# deviation: mu = 10 and sigma = 1 imply the same coefficient of variation, .10.
ss_aipe_cv(mu = 10, sigma = 1, width = .1, conf_level = .99)

# Ensuring that the confidence interval will be sufficiently narrow with a 99%
# certainty for the situation above.
ss_aipe_cv(C_of_V = .1, width = .1, conf_level = .99, assurance = .99)
```

`ss_aipe_cv_sensitivity`

Sensitivity Analysis for Sample Size Planning From the Accuracy in Parameter Estimation Perspective for the Coefficient of Variation

---

Description

Quantifies how much misspecification of the population coefficient of variation can distort an AIPE-based sample size plan. Given a true (population) value of the coefficient of variation and the value that was used in planning, the function simulates draws of size N from a normal population, computes the confidence interval for the coefficient of variation on each replication, and summarizes how often the realized interval width is below the desired target and how often the interval covers the population value. This is the standard sensitivity-analysis workflow described in Kelley (2007) and Maxwell, Delaney, and Kelley (2027, Section 3.11 on sample size planning).

Usage

```
ss_aipe_cv_sensitivity(
  true_cv = NULL,
  estimated_cv = NULL,
  width = NULL,
  assurance = NULL,
  mean = 100,
  specified_N = NULL,
  conf_level = 0.95,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

`true_cv`	Population coefficient of variation (the data generating value)
`estimated_cv`	Coefficient of variation used to plan the study (the value the researcher guessed when invoking `ss_aipe_cv`); must be positive. Supply this or `specified_N` but not both.
`width`	Desired (full) width of the two-sided confidence interval for the population coefficient of variation
`assurance`	Probability with which the realized interval should be no wider than `width` (must be `NULL` or strictly between 0 and 1; `NULL` means plan to the *expected* width without an assurance constraint)
`mean`	Population mean used by the simulator to generate data (the standard deviation is determined by `mean` and `true_cv`, since CV = sigma/mu). Default 100.
`specified_N`	Pre-specified sample size to evaluate (use this when you want the sensitivity results at a fixed N rather than at the N that `ss_aipe_cv` would recommend); incompatible with `estimated_cv`.
`conf_level`	Desired confidence level (i.e., 1 - Type I error rate); default 0.95
`G`	Number of Monte Carlo replications; defaults to 1000. Increase (e.g., 5000 or 10000) for stable Type I error estimates.
`print_iter`	Logical. If `TRUE` the simulation prints the iteration index after each replication (helpful for long runs); default `FALSE`.
`filename`	An optional path for a comma separated file recording every replication (the two confidence limits, the realized coefficient of variation, a coverage indicator, and the interval width): nothing is written when `filename` is `NULL` (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at `tempfile(fileext = ".csv")`.

Details

Sample size planning for the coefficient of variation under the Accuracy in Parameter Estimation framework chooses N so that the expected (or, with assurance, the high-probability) confidence

interval width is no larger than `width` (Kelley, 2007). Because the procedure assumes the planning value `estimated_cv` matches the population value `true_cv`, in practice the realized width will deviate from the planned width whenever the planning value is wrong. This sensitivity analysis quantifies the deviation by Monte Carlo simulation: the planned *N* is obtained from `ss_aipe_cv` with `estimated_cv`, then samples are drawn from the *true* population (with coefficient of variation `true_cv`) and the realized confidence interval widths are summarized.

For a discussion of AIPE-based sample size planning more generally and how sensitivity analyses guard against misspecification, see Maxwell, Delaney, & Kelley (2027, Section 3.5).

Value

A `data.frame` with columns `term` and `value` summarizing the Monte Carlo results across the `G` replications. The `term` entries are: `"mean_cv"`, `"median_cv"`, `"sd_cv"` (mean / median / SD of the *G* observed sample coefficients of variation); `"mean_ci_width"`, `"median_ci_width"`, `"sd_ci_width"` (corresponding summaries of the realized interval widths); `"pct_ci_less_w"` (proportion of intervals at or below the planning width `width`); `"pct_ci_miss_low"` and `"pct_ci_miss_high"` (tail-specific non-coverage); `"total_type_I_error"` (overall empirical non-coverage of `true_cv`); plus the input echoes `"total_N"` (the sample size evaluated), `"true_cv"`, `"estimated_cv"` (NA when `specified_N` was supplied instead), `"width"`, `"conf_level"`, and `"assurance"` (present only when an assurance was supplied). The proportion rows are on the 0 to 1 scale, not percentages, so `total_type_I_error` is the sum of `pct_ci_miss_low` and `pct_ci_miss_high`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Chattopadhyay, B., & Kelley, K. (2016). Estimation of the coefficient of variation with minimum risk: A sequential method for minimizing sampling error and study cost. *Multivariate Behavioral Research, 51*(5), 627–648. doi:10.1080/00273171.2016.1203279

Kelley, K. (2007). Sample size planning for the coefficient of variation from the accuracy in parameter estimation approach. *Behavior Research Methods, 39*(4), 755–766. doi:10.3758/BF03192966

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3.)

See Also

`ss_aipe_cv`, `ci_cv`, `cv`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Textbook scenario (Kelley, 2007). A researcher plans a study to estimate the
# coefficient of variation of reaction times with a 95% confidence interval no
# wider than .10. They guess from prior work that the population coefficient of
# variation is around .25 and apply ss_aipe_cv() to obtain a planned N.
```

```
# Question 1: how does the realized interval width behave when the planning
# value is correct (well-specified case)?
set.seed(113)
ss_aipe_cv_sensitivity(
  true_cv      = .25,
  estimated_cv = .25,
  width            = .10,
  assurance        = NULL,
  conf_level       = .95,
  G                = 200,
  print_iter       = FALSE
)

# Question 2: what happens if the planning value is materially smaller than
# the true coefficient of variation (a common direction of misspecification,
# since planning values are often optimistic)? The intervals will be wider on
# average than the target and the pct_ci_less_w will fall.
set.seed(113)
ss_aipe_cv_sensitivity(
  true_cv      = .35,
  estimated_cv = .25,
  width            = .10,
  assurance        = NULL,
  conf_level       = .95,
  G                = 200,
  print_iter       = FALSE
)
```

---

ss_aipe_equivalence_r *AIPE Sample Size Planning for an Equivalence Test on the Pearson Correlation*

Description

Computes the minimum sample size needed so that the equivalence CI (the 100(1 - 2α)% CI on the Pearson correlation ρ) has expected full width $\leq \omega$ (Kelley, 2007; Lakens, 2017). With the standard $\alpha = 0.05$ TOST level, the equivalence CI is the 90% CI. The interval is the Fisher's Z construction that `equivalence_r` and `ci_r` use, so the plan and the analysis invert the same interval.

Usage

```
ss_aipe_equivalence_r(
  population_r = 0,
  width,
  alpha_level = 0.05,
  assurance = NULL
)
```

Arguments

`population_r`	Anticipated population correlation ρ used to plan the width. Default `0`: the confidence interval for a correlation is widest at $\rho = 0$, so the default plans under the widest-interval case and is conservative for any other value.
`width`	Target full CI width on the correlation scale (e.g., `0.20` for a 90% CI of width 0.20). Must be in $(0, 2)$, the width of the correlation scale itself.
`alpha_level`	One-sided TOST significance level. The CI used in planning is at confidence level $1 - 2\alpha$. Default `0.05` (90% CI).
`assurance`	Optional assurance probability in $(0, 1)$. When supplied, the function chooses *N* so that the probability of achieving `width` or less is at least `assurance` (Kelley, Maxwell, & Rausch, 2003). Default `NULL` (no assurance correction).

Details

Closed form on the Fisher's Z scale. The equivalence CI has half-width $h = z_{1-\alpha}/\sqrt{N-3}$ on the Fisher's Z scale, and its width on the correlation scale is

$$w(N) \;=\; \tanh(Z_\rho + h) - \tanh(Z_\rho - h),$$

where $Z_\rho = \tanh^{-1}(\rho)$. The function returns the smallest integer $N \geq 4$ with $w(N) \leq \omega$. At $\rho = 0$ this is available in closed form, $N = \lceil 3 + (z_{1-\alpha}/\tanh^{-1}(\omega/2))^2 \rceil$, and away from zero the back-transform shortens the interval, so the required *N* can only decrease as $|\rho|$ grows.

Choosing the width from equivalence bounds. To leave room for an equivalence verdict inside bounds $(-b, b)$, the interval must at minimum fit inside the bounds when centered at the anticipated ρ, so a width somewhat below $2b$ (for ρ near 0) is the natural target; the Monte Carlo sensitivity sibling `ss_aipe_equivalence_r_sensitivity` reports the realized proportion of equivalence verdicts at the planned *N*.

Assurance. Under `assurance = q`, the function increments N until the Monte Carlo probability that the realized width is $\leq \omega$ is at least q, drawing the sampling distribution of $\widehat{Z}$ as normal with mean Z_ρ and variance $1/(N-3)$.

Value

A 4-row `data.frame` with columns `term` and `value`: the recommended sample size `necessary_N`, the target `width`, the planning value `population_r`, and the resulting `ci_width_expected` at the chosen *N*.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Counsell, A., & Cribbie, R. A. (2015). Equivalence tests for comparing correlation and regression coefficients. *British Journal of Mathematical and Statistical Psychology, 68*(2), 292–309. doi:10.1111/bmsp.12045

Goertzen, J. R., & Cribbie, R. A. (2010). Detecting a lack of association: An equivalence testing approach. *British Journal of Mathematical and Statistical Psychology, 63*(3), 527–537. doi:10.1348/000711009X475853

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., Maxwell, S. E., & Rausch, J. R. (2003). Obtaining power or obtaining precision: Delineating methods of sample size planning. *Evaluation and the Health Professions, 26*(3), 258–287. doi:10.1177/0163278703255242

Lakens, D. (2017). Equivalence tests: A practical primer for *t* tests, correlations, and meta-analyses. *Social Psychological and Personality Science, 8*(4), 355–362. doi:10.1177/1948550617697177

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics, 15*(6), 657–680.

See Also

equivalence_r, ss_aipe_r, ci_r, ss_aipe_equivalence_smd, ss_aipe_equivalence_r_sensitivity

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r_sensitivity(), ss_aipe_equivalence_smd(),
ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_icc_sensitivity(), ss_aipe_indirect_effec
ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(), ss_aipe_omega_squared(),
ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(), ss_aipe_partial_r_sensitivity(),
ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(), ss_aipe_reliability_sensitivity(),
ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# 1. Plan for a 90% CI on the correlation of width <= 0.20, under
#    the widest-interval planning value rho = 0:
ss_aipe_equivalence_r(population_r = 0, width = 0.20)

# 2. The same width assuming a true correlation of 0.30 requires
#    fewer participants, since the interval narrows away from zero:
ss_aipe_equivalence_r(population_r = 0.30, width = 0.20)

# 3. With 80% assurance (the assurance path is Monte Carlo, so seed
#    for a reproducible result):
set.seed(113)
ss_aipe_equivalence_r(population_r = 0.30, width = 0.20, assurance = 0.80)
```

`ss_aipe_equivalence_r_sensitivity`

Sensitivity Analysis for Sample Size Planning From the AIPE Perspective for an Equivalence-Test Correlation

---

Description

Quantifies how much misspecification of the population correlation distorts an AIPE-based sample size plan for the two-one-sided-tests (TOST) confidence interval on the Pearson correlation. On each replication the function simulates N bivariate normal pairs with population correlation `true_r`, computes the sample correlation and its Fisher's Z confidence interval via `ci_r`, and summarizes the realized widths and the proportion of replications in which the computed interval falls entirely inside (`equivalent`) the specified equivalence bounds.

Usage

```
ss_aipe_equivalence_r_sensitivity(
  true_r = 0,
  estimated_r = NULL,
  width,
  rho_lower = NULL,
  rho_upper = NULL,
  specified_N = NULL,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

`true_r`	Population correlation (the data generating value). Defaults to `0` (no association, the exact equivalence case).
`estimated_r`	Planning value of the population correlation passed to `ss_aipe_equivalence_r`; supply this or `specified_N` but not both.
`width`	Desired full width of the two-sided CI on the correlation.
`rho_lower, rho_upper`	Equivalence bounds on the correlation, as positive magnitudes with the same meaning as in `equivalence_r`: the region is $(-\rho_L, +\rho_U)$. `rho_upper` is required; `rho_lower` defaults to `rho_upper` (a symmetric region). The simulator records whether the realized CI falls entirely inside the region.
`specified_N`	Sample size to evaluate.
`conf_level`	Confidence level (default `0.95`).
`assurance`	Optional assurance probability.
`G`	Number of Monte Carlo replications.
`print_iter`	Logical.
`filename`	Optional path for a comma separated file recording every replication (the sample correlation, the two confidence limits, the interval width, whether the interval fell inside the equivalence region, and two indicators of whether the interval missed `true_r` below or above): nothing is written when `filename` is `NULL` (the default), a new file with a header row is created otherwise, an existing file at that

path is appended to, and a throwaway run should point it at `tempfile(fileext = ".csv")`.

Value

A `data.frame` with rows for mean / median / SD of the realized correlation and CI width, the proportion of intervals at or below `width`, tail-specific and overall non-coverage of `true_r`, the proportion of intervals classified as `equivalent` (CI fully inside the bounds), and the input echoes, including `assurance` (present only when an assurance was supplied).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Counsell, A., & Cribbie, R. A. (2015). Equivalence tests for comparing correlation and regression coefficients. *British Journal of Mathematical and Statistical Psychology, 68*(2), 292–309. doi:10.1111/bmsp.12045

Goertzen, J. R., & Cribbie, R. A. (2010). Detecting a lack of association: An equivalence testing approach. *British Journal of Mathematical and Statistical Psychology, 63*(3), 527–537. doi:10.1348/000711009X475853

See Also

ss_aipe_equivalence_r, equivalence_r, ss_aipe_r_sensitivity, ss_aipe_equivalence_smd_sensitivity

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_se
ss_aipe_icc(), ss_aipe_icc_sensitivity(), ss_aipe_indirect_effect(), ss_aipe_indirect_effect_sensitivi
ss_aipe_mixed_effects_sensitivity(), ss_aipe_omega_squared(), ss_aipe_omega_squared_sensitivity(),
ss_aipe_partial_r(), ss_aipe_partial_r_sensitivity(), ss_aipe_pcm_sensitivity(), ss_aipe_r(),
ss_aipe_r_sensitivity(), ss_aipe_reliability_sensitivity(), ss_aipe_semipartial_r(),
ss_aipe_semipartial_r_sensitivity()

Examples

```
# Reduced Monte Carlo sweep (small G) for a fast, illustrative run.
set.seed(113)
ss_aipe_equivalence_r_sensitivity(
  true_r      = 0.0,
  estimated_r = 0.0,
  width       = 0.30,
  rho_upper   = 0.20,
  G = 50, print_iter = FALSE
)
```

`ss_aipe_equivalence_smd`

AIPE Sample Size Planning for an Equivalence Test on the Standardized Mean Difference

Description

Computes the minimum per-group sample size needed so that the equivalence CI (the 100(1 - 2α)% CI on the standardized mean difference) has expected full width $\leq \omega$, that is, expected half-width $\leq \omega/2$ (Kelley, 2007; Lakens, 2017). With the standard $\alpha = 0.05$ TOST level, the equivalence CI is the 90% CI. The function inverts the large-sample variance of *d*; the noncentral *t* distribution of *d* enters through the optional assurance step.

Usage

```
ss_aipe_equivalence_smd(
  population_smd = 0,
  width,
  alpha_level = 0.05,
  assurance = NULL,
  balanced = TRUE
)
```

Arguments

`population_smd`	Anticipated population standardized mean difference δ used to plan the variance. Default `0`: plans under the most-conservative null-effect case.
`width`	Target full CI width on the *d* scale (e.g., `0.20` for a 90% CI of width 0.20).
`alpha_level`	One-sided TOST significance level. The CI used in planning is at confidence level $1 - 2\alpha$. Default `0.05` (90% CI).
`assurance`	Optional assurance probability in $(0, 1)$. When supplied, the function chooses *n* so that the probability of achieving `width` or less is at least `assurance` (Kelley, Maxwell, & Rausch, 2003). Default `NULL` (no assurance correction).
`balanced`	Logical; `TRUE` (default) plans equal-n groups. Unequal-n planning is not yet supported.

Details

Approximate-variance plan. The large-sample variance of *d* is

$$\mathrm{Var}(\hat{d}) \;\approx\; (n_1 + n_2)/(n_1 n_2) + d^2/(2(n_1 + n_2)).$$

For a balanced design with per-group size n, the half-width of the equivalence CI at level $1 - 2\alpha$ is approximately $z_{1-\alpha}\sqrt{\mathrm{Var}(\hat{d})}$. The function solves for the smallest integer n giving expected half-width $\leq \omega/2$.

Assurance. Under `assurance = q`, the function increments n until the simulated probability that the realized half-width is $\leq \omega/2$ is at least q. (Implemented as a thin Monte Carlo overlay. At the default planning value `population_smd = 0` the shift is typically zero; it grows with the planning value, reaching several per group by `population_smd = 0.5` with a narrow target width.)

Note on conservatism of the assurance plan. The empirical simulation study of the AIPE planner family finds that `ss_aipe_equivalence_smd()` is tight at $\gamma = 0.80$ but operates on the boundary of its valid range at $\gamma = 0.99$: the realized assurance at the recommended sample size is within Monte Carlo error of the target, typically a few tenths of a percentage point below 0.99. The mechanism is that the planner inverts a normal approximation to $\Pr(\widehat{W} > \omega)$, and at the 99% level the upper tail of $\widehat{W}$ is heavier than the approximation accounts for. Adding a small safety margin (5 to 10 subjects per group) restores the desired probability statement when planning at high assurance; `ss_aipe_equivalence_smd_sensitivity` reproduces the check for any one condition.

Value

A 5-row `data.frame` with columns `term` and `value`: the per-group recommended sample size `necessary_n_per_group`, the implied total `total_N`, the target `width`, the planning value `population_smd`, and `ci_width_expected`, the expected full CI width at the chosen *n*.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Kelley, K., Maxwell, S. E., & Rausch, J. R. (2003). Obtaining power or obtaining precision: Delineating methods of sample size planning. *Evaluation and the Health Professions, 26*(3), 258–287. doi:10.1177/0163278703255242

Lakens, D. (2017). Equivalence tests: A practical primer for *t* tests, correlations, and meta-analyses. *Social Psychological and Personality Science, 8*(4), 355–362. doi:10.1177/1948550617697177

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons and Chapter 3 on one-way ANOVA.)

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics, 15*(6), 657–680.

See Also

`equivalence_smd`, `ss_aipe_smd`, `ci_smd`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(),
ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_icc_sensitivity(), ss_aipe_indirect_effec
ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(), ss_aipe_omega_squared(),
ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(), ss_aipe_partial_r_sensitivity(),
ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(), ss_aipe_reliability_sensitivity(),
ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# 1. Plan for a 90% CI on d of width <= 0.20, under d_planning = 0:
ss_aipe_equivalence_smd(population_smd = 0, width = 0.20)

# 2. Plan for the same width assuming a true d = 0.05:
ss_aipe_equivalence_smd(population_smd = 0.05, width = 0.20)

# 3. With 80% assurance (the assurance path is Monte Carlo, so seed for
#    a reproducible result):
set.seed(113)
ss_aipe_equivalence_smd(population_smd = 0.05, width = 0.20, assurance = 0.80)
```

---

ss_aipe_equivalence_smd_sensitivity

Sensitivity Analysis for Sample Size Planning From the AIPE Perspective for an Equivalence-Test SMD

Description

Quantifies how much misspecification of the population standardized mean difference distorts an AIPE-based sample size plan for the two-one-sided-tests (TOST) confidence interval on the SMD. On each replication the function simulates two normal groups of size n per group with population standardized mean difference `true_smd`, computes the SMD and its noncentral t confidence interval via `ci_smd`, and summarizes the realized widths and the proportion of replications in which the computed interval falls entirely inside (`equivalent`) the specified equivalence bounds.

Usage

```
ss_aipe_equivalence_smd_sensitivity(
  true_smd = 0,
  estimated_smd = NULL,
  width,
  delta_lower = NULL,
  delta_upper = NULL,
  n_per_group = NULL,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
```

```
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

`true_smd`	Population standardized mean difference (the data generating value). Defaults to `0` (perfect equivalence).
`estimated_smd`	Planning value of the population SMD passed to `ss_aipe_equivalence_smd`; supply this or `n_per_group` but not both.
`width`	Desired full width of the two-sided CI on the SMD.
`delta_lower, delta_upper`	Equivalence bounds on the SMD, as positive magnitudes with the same meaning as in `equivalence_smd`: the region is $(-\delta_L, +\delta_U)$, with `delta_lower` as δ_L and `delta_upper` as δ_U. `delta_upper` is required; `delta_lower` defaults to `delta_upper` (a symmetric region). The simulator records whether the realized CI falls entirely inside the region.
`n_per_group`	Per-group sample size to evaluate.
`conf_level`	Confidence level (default `0.95`).
`assurance`	Optional assurance probability.
`G`	Number of Monte Carlo replications.
`print_iter`	Logical.
`filename`	Optional path for a comma separated file recording every replication (the sample standardized mean difference, the two confidence limits, the interval width, whether the interval fell inside the equivalence region, and two indicators of whether the interval missed `true_smd` below or above): nothing is written when `filename` is `NULL` (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at `tempfile(fileext = ".csv")`.

Value

A `data.frame` with rows for mean / median / SD of the realized SMD and CI width, the proportion of intervals at or below `width`, tail-specific and overall non-coverage of `true_smd`, the proportion of intervals classified as `equivalent` (CI fully inside the bounds), and the input echoes, including `assurance` (present only when an assurance was supplied).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*, 363–385. doi:10.1037/1082989X.11.4.363

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

ss_aipe_equivalence_smd, equivalence_smd, ss_aipe_smd_sensitivity

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(), ss_aipe_equivalence_smd(), ss_aipe_icc(), ss_aipe_icc_sensitivity(), ss_aipe_indirect_effect(), ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(), ss_aipe_omega_squared(), ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(), ss_aipe_partial_r_sensitivity(), ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(), ss_aipe_reliability_sensitivity(), ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# Reduced Monte Carlo sweep (small G) for a fast, illustrative run.
set.seed(113)
ss_aipe_equivalence_smd_sensitivity(
  true_smd      = 0.0,
  estimated_smd = 0.0,
  width         = 0.30,
  delta_upper   = 0.20,
  G = 50, print_iter = FALSE
)
```

---

ss_aipe_icc *Sample Size for AIPE on an Intraclass Correlation Coefficient*

Description

Determines the sample size needed for a confidence interval on a population intraclass correlation coefficient (ICC) to have a desired width, using Bonett's (2002) Fisher-style variance-stabilizing transformation. The function inverts the asymptotic variance on the transformed scale (where the CI is symmetric and approximately normal), solves for the smallest n that achieves the target half-width on the back-transformed (raw-ICC) scale, and optionally inflates the result by a chi squared assurance correction (Kelley & Maxwell, 2003).

Usage

```
ss_aipe_icc(
  rho,
  k,
  width,
  which_width = c("Full", "Lower", "Upper"),
  conf_level = 0.95,
 type = c("ICC(1,1)", "ICC(2,1)", "ICC(3,1)", "ICC(1,k)", "ICC(2,k)", "ICC(3,k)"),
  assurance = NULL
)
```

Arguments

rho	Anticipated population ICC at the level matching type, in $[0, 1)$. This is the value the researcher expects the truth to be near, motivated by prior literature, a pilot, or substantive theory. Because the planning formula inverts the asymptotic variance *at* this value, a wrong guess inflates or deflates the realized confidence interval width relative to width; the function ss_aipe_icc_sensitivity quantifies the impact of misspecification by Monte Carlo.
k	Number of raters (or measurements per subject); must be at least 2.
width	Desired full width of the back-transformed CI on the ICC.
which_width	Whether width is the "Full" width of the interval (default) or a half-width: "Lower" and "Upper" both interpret width as half the full width, so they plan for a full width of twice width and return the same sample size. Because the interval is not generally symmetric about the estimate, its realized lower and upper half-widths can differ from each other and from half the full width; the planner does not target them separately. A genuinely one-sided width target is not currently offered.
conf_level	Desired confidence level (default 0.95).
type	Which Shrout-Fleiss (1979) ICC form is being planned. One of the single-rater forms "ICC(1,1)", "ICC(2,1)", "ICC(3,1)" (which share the planning variance) or the average-of-k forms "ICC(1,k)", "ICC(2,k)", "ICC(3,k)"; default "ICC(1,1)". Both rho and width are interpreted on the scale of the requested form: for an average-of-k form the planning value is mapped to the single-rater scale through the inverse Spearman-Brown relation and each candidate confidence limit is mapped back, so the returned sample size targets the width of the interval on the average-of-k ICC itself (see Details).
assurance	Optional. Probability that the realized CI is no wider than width; when supplied, the sample size is inflated by the standard chi squared correction.

Details

Bonett's (2002) Fisher-style transform. Bonett (2002) showed that the transformation

$$L(\rho) \;=\; \frac{1}{2}\log\!\left(\frac{1+(k-1)\rho}{1-\rho}\right)$$

approximately variance-stabilizes the single-rater ICC, with

$$\mathrm{Var}(L(\hat{\rho})) \;\approx\; \frac{k}{2\,(k-1)\,(n-2)}.$$

A confidence interval is constructed by adding $\pm z_{1-\alpha/2}$ standard errors on the L scale and back-transforming to the raw-ICC scale via $\rho = (e^{2L}-1)/(e^{2L}-1+k)$. The minimum sample size is found by searching for the smallest n whose back-transformed CI width is below the target.

Single-rater vs.\ average-of-k ICC. The Bonett (2002) variance applies directly to the single-rater forms (ICC(1,1), ICC(2,1), ICC(3,1)). For the average-of-k forms the planning value ρ_k is first mapped to the single-rater scale through the inverse Spearman-Brown relation $\rho = \rho_k/[k-(k-1)\rho_k]$, the interval is formed on the L scale as above, and each candidate limit is mapped back

to the average-of-k scale (composing the inverse L transform with the Spearman-Brown formula reduces to $\rho_k = 1 - e^{-2L}$). The width criterion therefore applies to the confidence interval on the average-of-k ICC itself, following the convention used by `var_icc`. Because the two scales differ, an average-of-k plan generally recommends a different sample size than a single-rater plan at the same numeric `rho` and `width`; at `rho = 0.7`, `k = 3`, and `width = 0.20`, planning for `ICC(1,1)` recommends $n = 69$ subjects while planning for `ICC(1,k)` recommends $n = 110$.

The assurance correction can fall slightly short. Monte Carlo evaluation with `ss_aipe_icc_sensitivity` shows that the chi squared inflation tends to deliver a little less assurance than requested. At the condition of the second example below (`rho = 0.7`, `k = 3`, `width = 0.20`, `assurance = 0.80`), the recommended $n = 79$ yields an empirical assurance of about .77 against the requested .80 (10,000 replications of the F-based interval computed by `icc`), and the smallest sample size whose empirical assurance reaches .80 is $n = 81$. The mechanism is that the realized interval widths on the raw-ICC scale have a heavier upper tail than the chi squared inflation on the transformed scale accounts for, so the buffer the correction adds is slightly too small. When meeting the assurance target matters, check the recommended sample size with `ss_aipe_icc_sensitivity` and increase n until the empirical assurance reaches the target.

Value

A `data.frame` with rows for the recommended sample size (*number of subjects*), the expected back-transformed CI width, and the inputs echoed back. The Shrout-Fleiss form the plan targets is stored as the `"icc_type"` attribute so the `value` column stays numeric.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bonett, D. G. (2002). Sample size requirements for estimating intraclass correlations with desired precision. *Statistics in Medicine, 21*(9), 1331–1335. doi:10.1002/sim.1108

Donner, A. (1986). A review of inference procedures for the intraclass correlation coefficient in the one-way random effects model. *International Statistical Review, 54*(1), 67–82.

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Shrout, P. E., & Fleiss, J. L. (1979). Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin, 86*(2), 420–428.

Smith, C. A. B. (1956). On the estimation of intraclass correlation. *Annals of Human Genetics, 21*(4), 363–373.

See Also

`icc`, `var_icc`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(),
ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc_sensitivity(),
ss_aipe_indirect_effect(), ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(),
ss_aipe_omega_squared(), ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(),
ss_aipe_partial_r_sensitivity(), ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(),
ss_aipe_reliability_sensitivity(), ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# 1. Plan n so the 95% CI on a single-rater ICC has full width <= 0.20
#        with k = 3 raters and an anticipated ICC of 0.7.
ss_aipe_icc(rho = 0.7, k = 3, width = 0.20)

# 2. With 80% assurance:
ss_aipe_icc(rho = 0.7, k = 3, width = 0.20, assurance = 0.80)
```

`ss_aipe_icc_sensitivity` *Sensitivity Analysis for Sample Size Planning From the Accuracy in Parameter Estimation Perspective for an Intraclass Correlation Coefficient*

Description

Quantifies how much misspecification of the population ICC can distort an AIPE-based sample size plan. Given a true (population) ICC and the value used in planning, the function simulates draws of size $n \times k$ from the relevant variance-components model, computes the ICC and its F-distribution confidence interval on each replication via icc, and summarizes how often the realized interval width is below the desired target and how often the interval covers the population value. This is the standard sensitivity-analysis workflow described in Kelley (2007) and Maxwell, Delaney, and Kelley (2027, Section 3.11 on sample size planning).

Usage

```
ss_aipe_icc_sensitivity(
  true_rho = NULL,
  estimated_rho = NULL,
  k,
  width,
  assurance = NULL,
  specified_N = NULL,
  conf_level = 0.95,
 type = c("ICC(1,1)", "ICC(2,1)", "ICC(3,1)", "ICC(1,k)", "ICC(2,k)", "ICC(3,k)"),
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

true_rho	Population intraclass correlation coefficient (the data generating value), at the level matching type. Must lie in $[0, 1)$.
estimated_rho	ICC used to plan the study (the value the researcher guessed when invoking ss_aipe_icc). Supply this or specified_N but not both.
k	Number of raters (or repeated measurements) per subject; must be at least 2.
width	Desired full width of the (back-transformed) confidence interval on the ICC.
assurance	Probability with which the realized interval should be no wider than width (must be NULL or strictly between 0 and 1; NULL means plan to the *expected* width without an assurance constraint).
specified_N	Pre-specified number of subjects to evaluate (use this when you want the sensitivity results at a fixed n rather than at the n that ss_aipe_icc would recommend); incompatible with estimated_rho.
conf_level	Desired confidence level (i.e., 1 minus the Type I error rate); default 0.95.
type	Which Shrout-Fleiss (1979) ICC form is being planned. One of "ICC(1,1)" (default; one-way random model), "ICC(2,1)" (two-way random model, absolute agreement), "ICC(3,1)" (two-way mixed model, consistency), or the average-of-k versions "ICC(1,k)", "ICC(2,k)", "ICC(3,k)". The simulation generates data from the variance- components model matching the requested type, so the realized ICC on each replication is comparable to true_rho.
G	Number of Monte Carlo replications; defaults to 1000. Increase (e.g., 5000 or 10000) for stable empirical-coverage estimates.
print_iter	Logical. If TRUE the simulation prints the iteration index after each replication (helpful for long runs); default FALSE.
filename	An optional path for a comma separated file recording every replication (the realized ICC, the two confidence limits, the interval width, and the two tail-specific non-coverage indicators): nothing is written when filename is NULL (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at tempfile(fileext = ".csv").

Details

Sample size planning for the intraclass correlation coefficient under the Accuracy in Parameter Estimation framework chooses n so that the expected (or, with assurance, the high-probability) confidence interval width is no larger than width (Bonett, 2002; Kelley & Maxwell, 2003). Because the procedure assumes the planning value estimated_rho matches the population value true_rho, in practice the realized width will deviate from the planned width whenever the planning value is wrong. This sensitivity analysis quantifies the deviation by Monte Carlo simulation: the planned n is obtained from ss_aipe_icc with estimated_rho, then samples are drawn from the *true* population (with population ICC true_rho) and the realized confidence interval widths are summarized.

Data generating model. For type starting with "ICC(1," the simulation uses the one-way random model: each row (subject) gets a subject random effect, every cell adds independent Gaussian noise, and the population ICC equals $\sigma^2_{\text{subj}}/(\sigma^2_{\text{subj}}+\sigma^2_{\text{err}})$. For type starting with "ICC(2," the simulation also adds a rater random effect, so the population ICC(2,1) equals $\sigma^2_{\text{subj}}/(\sigma^2_{\text{subj}} + \sigma^2_{\text{rater}} + \sigma^2_{\text{err}})$.

For type starting with "ICC(3," the rater effect is fixed (centered constants), so the population ICC(3,1) equals $\sigma^2_{\text{subj}}/(\sigma^2_{\text{subj}} + \sigma^2_{\text{err}})$ but the two-way decomposition is used in the estimator. Within each cell, the total variance is unity by construction. For the single-rater forms true_rho therefore maps directly to the subject-variance share; for the average-of-k forms true_rho is first mapped to the single-rater scale through the inverse Spearman-Brown relation $\rho = \rho_k/[k - (k - 1)\rho_k]$, so that the population ICC at the average-of-k level equals true_rho. Coverage is checked against true_rho on its own scale, matching the scale on which icc reports each form.

Value

A data.frame with columns term and value summarizing the Monte Carlo results across the G replications. The term entries are: "mean_icc", "median_icc", "sd_icc" (mean / median / SD of the G observed ICC estimates); "mean_ci_width", "median_ci_width", "sd_ci_width" (corresponding summaries of the realized interval widths); "pct_ci_less_w" (proportion of intervals at or below the planning width width); "pct_ci_miss_low" and "pct_ci_miss_high" (tail-specific non-coverage of true_rho); "total_type_I_error" (overall empirical non-coverage of true_rho); plus the input echoes "total_N", "k", "true_rho", "estimated_rho", "width", "conf_level", and "assurance" (present only when an assurance was supplied). The ICC type is not a row; it is stored as the "icc_type" attribute on the returned object so the value column stays numeric.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bonett, D. G. (2002). Sample size requirements for estimating intraclass correlations with desired precision. *Statistics in Medicine, 21*(9), 1331–1335. doi:10.1002/sim.1108

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapters 10 and 11 on intraclass correlation and reliability.)

Shrout, P. E., & Fleiss, J. L. (1979). Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin, 86*(2), 420–428.

See Also

ss_aipe_icc, icc, var_icc

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(),

ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_indirect_effec
ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(), ss_aipe_omega_squared(),
ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(), ss_aipe_partial_r_sensitivity(),
ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(), ss_aipe_reliability_sensitivity(),
ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# Reduced G and a wide target width keep this Monte Carlo example fast;
# raise G (e.g., 1000 or more) for stable empirical-coverage estimates.

# Well-specified case: plan with the true population ICC.
set.seed(113)
ss_aipe_icc_sensitivity(
  true_rho      = 0.70,
  estimated_rho = 0.70,
  k             = 3,
  width         = 0.40,
  conf_level    = 0.95,
  G             = 25,
  print_iter    = FALSE
)

# Misspecified case: the planner used .70 but the truth is .50.
# The realized interval widths will tend to be wider than the target.
set.seed(113)
ss_aipe_icc_sensitivity(
  true_rho      = 0.50,
  estimated_rho = 0.70,
  k             = 3,
  width         = 0.40,
  conf_level    = 0.95,
  G             = 25,
  print_iter    = FALSE
)

# Fixed-n mode: skip the planner and evaluate at a chosen sample size.
set.seed(113)
ss_aipe_icc_sensitivity(
  true_rho    = 0.50,
  specified_N = 40,
  k           = 3,
  width       = 0.40,
  G           = 25,
  print_iter  = FALSE
)
```

---

ss_aipe_indirect_effect

Sample Size for AIPE on a Mediated (Indirect) Effect ab

Description

Determines the sample size needed for the confidence interval on a mediated effect ab (the product of the $X \rightarrow M$ and $M \rightarrow Y$ coefficients in a simple three-variable mediation model) to have a desired full width. Two methods are available. `"closed_form"` (the default) plans for the symmetric Wald interval built on the delta method standard error of the product (Sobel, 1982) and answers instantly. `"monte_carlo"` plans for the Monte Carlo confidence interval (MacKinnon, Lockwood, & Williams, 2004; Tofighi & MacKinnon, 2011), the interval `mediation_mbco` reports under `ci_method = "monte_carlo"`, and it plans by a priori Monte Carlo simulation: at each candidate sample size the mediation model is fit to `G` simulated data sets and the realized interval widths are recorded. Because the Monte Carlo interval respects the skewness of the sampling distribution of a product, and because the simulation measures the widths that fitted models actually deliver, `method = "monte_carlo"` is the recommended way to settle on the final sample size; the closed form is its fast first approximation.

Usage

```
ss_aipe_indirect_effect(
  a,
  b,
  width,
  method = c("closed_form", "monte_carlo"),
  conf_level = 0.95,
  n_max = 10000L,
  B = 5000L,
  G = 1000L,
  seed = NULL
)
```

Arguments

`a`	Anticipated population coefficient for $X \rightarrow M$, on the standardized scale. Numeric scalar in $(-1, 1)$.
`b`	Anticipated population coefficient for $M \rightarrow Y$ controlling for X, standardized. Numeric scalar in $(-1, 1)$.
`width`	Desired full width of the confidence interval on ab.
`method`	One of `"closed_form"` (default) or `"monte_carlo"`; see Details.
`conf_level`	Desired confidence level (default `0.95`).
`n_max`	Upper bound on the search; default `10000`.
`B`	Number of Monte Carlo draws forming the interval within each simulated study when `method = "monte_carlo"`; default `5000`. This is the same `B` the analysis-stage interval uses (see `mediation_mbco`).
`G`	Number of simulated studies per candidate sample size when `method = "monte_carlo"`; default `1000`. The mean simulated width carries a simulation error of about its standard deviation over $\sqrt{G}$; raise `G` for a sharper answer.
`seed`	Optional integer seed for the Monte Carlo method, used locally (the caller's random number generator state is restored on exit). Default `NULL` leaves the random number generator state alone.

Details

The mediation model. The simple mediator model is

$$M = \alpha_1 + aX + \varepsilon_M,$$

$$Y = \alpha_2 + c'X + bM + \varepsilon_Y,$$

with the indirect (mediated) effect of X on Y through M equal to ab (MacKinnon, Lockwood, Hoffman, West, & Sheets, 2002). Both methods plan on the standardized scale with no direct effect: the planning population takes X, M, and Y with unit variances and $c' = 0$, so `a` and `b` are the standardized paths.

The closed form. Under the planning population the sampling variance of $\hat{a}$ is $(1 - a^2)/(n - 2)$. In the equation for Y the mediator is regressed alongside X, with which it is correlated at a, so the sampling variance of $\hat{b}$ carries the variance inflation factor $1/(1 - a^2)$:

$$\mathrm{Var}(\hat{b}) \;=\; \frac{1 - b^2}{(n - 3)(1 - a^2)}.$$

The estimators come from two separate equations and are uncorrelated, so the delta method (Sobel, 1982) standard error of the product is

$$\mathrm{SE}(\hat{a}\hat{b}) \;=\; \sqrt{a^2\mathrm{Var}(\hat{b}) + b^2\mathrm{Var}(\hat{a})}\,,$$

and the closed form returns the smallest n at which the Wald width $2z_{1-\alpha/2}\,\mathrm{SE}(\hat{a}\hat{b})$ is at or below `width`. Two approximations remain. The Wald interval is symmetric while the sampling distribution of a product is skewed, so the Wald interval is not the interval an indirect effect should be reported with (Tofighi & Kelley, 2020). And the closed form evaluates the standard error at the planning values, while a fitted model evaluates it at the estimates, which leaves a discrepancy of a percent or two in realized width at moderate sample sizes. Both are reasons to treat the closed form as the first approximation and to verify the final plan with `method = "monte_carlo"`, which measures the realized widths directly.

Planning for the Monte Carlo interval. With `method = "monte_carlo"`, each candidate n is evaluated by a priori Monte Carlo simulation (Muthén & Muthén, 2002; Schoemann, Boulton, & Short, 2017): `G` data sets of size n are drawn from the planning population, the two mediation regressions are fit to each, and the Monte Carlo interval is formed by drawing `B` pairs $(\tilde{a}, \tilde{b})$ from normal distributions centered at the estimates with the estimated standard errors, multiplying, and reading off the empirical $(\alpha/2, 1 - \alpha/2)$ quantiles (MacKinnon, Lockwood, & Williams, 2004). Since $\hat{a}$ and $\hat{b}$ are uncorrelated here, the independent draws realize the joint normal approximation of the estimates, the same construction `mediation_mbco` uses for its Monte Carlo interval. The necessary sample size is the smallest n whose mean simulated width is at or below `width`; the search starts from the closed-form answer, brackets the crossing geometrically, and bisects. A planning call at the default `G` and `B` fits the mediation model several thousand times and takes a few seconds; the Monte Carlo example below lowers both to keep the page quick. The necessary sample size inherits the simulation error of the mean widths; raising `G` narrows it, and supplying `seed` makes a plan reproducible.

Relation to the MBCO procedure. The model-based constrained optimization (MBCO) likelihood ratio test of Tofighi and Kelley (2020), implemented in `mediation_mbco`, is the recommended test of a mediation effect, and the intervals that suit an indirect effect are the profile likelihood interval

and the Monte Carlo interval, both of which accommodate the skewness of the product. This planner targets the Monte Carlo interval. Planning for the profile likelihood interval would require inverting a pair of constrained optimizations in every simulated study (two constrained **OpenMx** fits per interval, times `G`, times every candidate sample size), while the Monte Carlo interval costs `B` products of normal draws per study and is the inexpensive interval that also accommodates the skewness, the one Tofighi and Kelley (2020) report for their memory example. A study planned with `method = "monte_carlo"` and analyzed with `mediation_mbco(ci_method = "monte_carlo")` is therefore planned and analyzed on the same interval.

Beyond the simple model. The planning population assumes standardized observed variables, one mediator, no covariates, and no direct effect. With a nonzero direct effect the residual variance of Y is $1 - b^2 - c'^2 - 2abc'$ rather than $1 - b^2$, so assuming $c' = 0$ errs toward a larger sample whenever $c'(c' + 2ab) > 0$ (consistent mediation) and toward a smaller one otherwise. When the direct effect, covariates, several mediators, or latent variables matter to the design, plan by simulation from the full model with `ss_aipe_composite_sem`, labeling the paths and defining the indirect effect via `ab := a*b`; its intervals are the Wald intervals of the fitted model, the same target as the closed form here. `ss_aipe_indirect_effect_sensitivity` quantifies what a plan from this page delivers when the population paths differ from the planning values.

Value

A `data.frame` with rows for the recommended sample size, the expected CI width at that size, and the inputs echoed back. Under `method = "closed_form"` the expected width is the delta method width evaluated at the returned sample size; under `method = "monte_carlo"` it is the mean simulated width there. The method is carried on the returned object as the `ci_method` attribute.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Fritz, M. S., & MacKinnon, D. P. (2007). Required sample size to detect the mediated effect. *Psychological Science, 18*(3), 233–239. doi:10.1111/j.14679280.2007.01882.x

Lachowicz, M. J., Preacher, K. J., & Kelley, K. (2018). A novel measure of effect size for mediation analysis. *Psychological Methods, 23*, 244–261. doi:10.1037/met0000165

MacKinnon, D. P., Lockwood, C. M., Hoffman, J. M., West, S. G., & Sheets, V. (2002). A comparison of methods to test mediation and other intervening variable effects. *Psychological Methods, 7*(1), 83–104. doi:10.1037/1082989X.7.1.83

MacKinnon, D. P., Lockwood, C. M., & Williams, J. (2004). Confidence limits for the indirect effect: Distribution of the product and resampling methods. *Multivariate Behavioral Research, 39*(1), 99–128. doi:10.1207/s15327906mbr3901_4

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Muthén, L. K., & Muthén, B. O. (2002). How to use a Monte Carlo study to decide on sample size and determine power. *Structural Equation Modeling, 9*(4), 599–620. doi:10.1207/S15328007SEM0904_8

Preacher, K. J., & Kelley, K. (2011). Effect size measures for mediation models: Quantitative strategies for communicating indirect effects. *Psychological Methods, 16*(2), 93–115. doi:10.1037/a0022658

Schoemann, A. M., Boulton, A. J., & Short, S. D. (2017). Determining power and sample size for simple and complex mediation models. *Social Psychological and Personality Science, 8*(4), 379–386. doi:10.1177/1948550617715068

Sobel, M. E. (1982). Asymptotic confidence intervals for indirect effects in structural equation models. *Sociological Methodology, 13*, 290–312.

Tofighi, D., & Kelley, K. (2020). Improved inference in mediation analysis: Introducing the model-based constrained optimization procedure. *Psychological Methods, 25*, 496–515. doi:10.1037/met0000259

Tofighi, D., & MacKinnon, D. P. (2011). RMediation: An R package for mediation analysis confidence intervals. *Behavior Research Methods, 43*(3), 692–700. doi:10.3758/s134280110076x

See Also

`mediation_mbco` for the analysis the plan feeds; `ss_aipe_composite_sem` for AIPE planning of an indirect effect in an arbitrary lavaan model; `ss_aipe_indirect_effect_sensitivity`; `ss_aipe_partial_r`, `ss_aipe_semipartial_r`, `ss_aipe_rc`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: `ss_aipe_c_sensitivity()`, `ss_aipe_cliff_delta()`, `ss_aipe_cliff_delta_sensit` `ss_aipe_composite_sem()`, `ss_aipe_equivalence_r()`, `ss_aipe_equivalence_r_sensitivity()`, `ss_aipe_equivalence_smd()`, `ss_aipe_equivalence_smd_sensitivity()`, `ss_aipe_icc()`, `ss_aipe_icc_sensitivit` `ss_aipe_indirect_effect_sensitivity()`, `ss_aipe_mixed_effects_sensitivity()`, `ss_aipe_omega_squared()`, `ss_aipe_omega_squared_sensitivity()`, `ss_aipe_partial_r()`, `ss_aipe_partial_r_sensitivity()`, `ss_aipe_pcm_sensitivity()`, `ss_aipe_r()`, `ss_aipe_r_sensitivity()`, `ss_aipe_reliability_sensitivity()`, `ss_aipe_semipartial_r()`, `ss_aipe_semipartial_r_sensitivity()`

Examples

```
# 1. Plan n so the 95% CI on ab has full width <= 0.20, with
#        anticipated standardized a = 0.40 and b = 0.40. The closed
#        form answers instantly:
ss_aipe_indirect_effect(a = 0.40, b = 0.40, width = 0.20)

# 2. The recommended plan targets the Monte Carlo interval directly:
#        every candidate sample size fits the mediation model to G
#        simulated data sets and measures the realized widths. G = 100
#        and B = 1000 keep the example quick; a reported plan deserves
#        the defaults G = 1000 and B = 5000. The answer sits a little
#        above the closed form because the interval it plans for is a
#        little wider than the Wald interval:
ss_aipe_indirect_effect(a = 0.40, b = 0.40, width = 0.20,
                        method = "monte_carlo", G = 100, B = 1000,
                        seed = 113)
```

`ss_aipe_indirect_effect_sensitivity` *Sensitivity Analysis for Sample Size Planning From the AIPE Perspective for an Indirect Effect*

Description

Quantifies how much misspecification of the population mediation path coefficients a and b distorts an AIPE-based sample size plan for the indirect effect ab. On each replication the function simulates a three-variable mediation system $X \rightarrow M \rightarrow Y$ of size n with population path coefficients `true_a` and `true_b`, fits the two regressions of M on X and Y on M and X, computes the sample indirect effect $\hat{a}\hat{b}$, and forms the interval the plan targeted: the symmetric Wald interval from the delta method standard error (`method = "closed_form"`) or the Monte Carlo interval (`method = "monte_carlo"`), matching `ss_aipe_indirect_effect`.

Usage

```
ss_aipe_indirect_effect_sensitivity(
  true_a = NULL,
  true_b = NULL,
  estimated_a = NULL,
  estimated_b = NULL,
  width,
  specified_N = NULL,
  method = c("closed_form", "monte_carlo"),
  conf_level = 0.95,
  B = 5000L,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

`true_a`	Population path coefficient a (from X to M); the data generating value.
`true_b`	Population path coefficient b (from M to Y after controlling for X); the data generating value.
`estimated_a, estimated_b`	Path coefficients used to plan the study (passed to `ss_aipe_indirect_effect`). Supply both or neither (if neither, supply `specified_N`).
`width`	Desired full width of the CI on ab.
`specified_N`	Sample size to evaluate (incompatible with `estimated_a` / `estimated_b`).
`method`	One of `"closed_form"` (default) or `"monte_carlo"`; the interval computed on each replication, also forwarded to the planner when the sample size is planned from `estimated_a` and `estimated_b`. A planning call with `method`

	= "monte_carlo" runs the planner's a priori Monte Carlo search at its default G, so it takes a few seconds.
conf_level	Confidence level (default 0.95).
B	Number of Monte Carlo draws used for the indirect-effect CI when method = "monte_carlo" (default 5000).
G	Number of outer simulation replications (default 1000).
print_iter	Logical.
filename	Optional path for a comma separated file recording every replication (the sample indirect effect $\hat{a}\hat{b}$, the two confidence limits, the interval width, and two indicators of whether the interval missed true_a * true_b below or above): nothing is written when filename is NULL (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at tempfile(fileext = ".csv").

Value

A data.frame with rows for mean / median / SD of $\hat{a}\hat{b}$ and the CI width, the proportion of intervals at or below width, tail-specific and overall non-coverage of the population value true_a * true_b, and the input echoes.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Preacher, K. J., & Kelley, K. (2011). Effect size measures for mediation models: Quantitative strategies for communicating indirect effects. *Psychological Methods, 16*(2), 93–115. doi:10.1037/a0022658

Tofighi, D., & Kelley, K. (2020). Improved inference in mediation analysis: Introducing the model-based constrained optimization procedure. *Psychological Methods, 25*, 496–515. doi:10.1037/met0000259

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

ss_aipe_indirect_effect, var_indirect_effect

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(),
ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_icc_sensitivi
ss_aipe_indirect_effect(), ss_aipe_mixed_effects_sensitivity(), ss_aipe_omega_squared(),
ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(), ss_aipe_partial_r_sensitivity(),
ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(), ss_aipe_reliability_sensitivity(),
ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# Reduced replications and a wide target width keep this fast.
set.seed(113)
ss_aipe_indirect_effect_sensitivity(
  true_a = 0.4, true_b = 0.3,
  estimated_a = 0.4, estimated_b = 0.3,
  width = 0.40, method = "closed_form",
  G = 50, print_iter = FALSE
)
```

---

ss_aipe_mixed_effects *AIPE Sample Size Planning for a Fixed Effect in a Two-Level Mixed-Effects Model*

Description

Computes the minimum number of clusters (level-2 units) needed so that the confidence interval on a level-1 fixed-effect slope has expected full width no larger than ω (Kelley, 2007; Raudenbush & Liu, 2001; Snijders & Bosker, 2012). The function inverts the closed-form approximation for the variance of a fixed-effect slope in a balanced two-level random-intercept model.

Usage

```
ss_aipe_mixed_effects(
  sigma2_y,
  sigma2_x,
  icc,
  width,
  cluster_size = 20L,
  conf_level = 0.95
)
```

Arguments

`sigma2_y`	Total variance of the outcome variable.
`sigma2_x`	Variance of the level-1 predictor (covariate).
`icc`	Intraclass correlation of the outcome.
`width`	Target full CI width on the slope.
`cluster_size`	Per-cluster sample size (number of level-1 units per level-2 unit). Default `20L`.
`conf_level`	Confidence level. Default `0.95`.

Details

Variance of the slope. For a level-1 predictor centered within cluster, the asymptotic variance of $\hat{\beta}$ is approximately

$$\mathrm{Var}(\hat{\beta}) \approx \frac{\sigma_y^2(1-\rho_I)}{N\sigma_x^2},$$

where $N = n_{\text{clusters}} \cdot m$ is the total number of level-1 units, m is the cluster size, and ρ_I the intraclass correlation. Because within-cluster centering removes the cluster-level variation from the predictor, the design effect $1+(m-1)\rho_I$ that inflates the variance of a cluster-level estimand does not appear here; clustering enters only through the residual variance $\sigma_y^2(1-\rho_I)$. The function inverts this expression for N. No anticipated slope value is needed: β does not appear in the variance, so the recommended number of clusters is the same whatever the slope.

Scope. Planning is for the most common single-level covariate case (random intercept, fixed slope, level-1 predictor centered within cluster). For cross-level interactions or random slopes, the variance formula changes and a Monte Carlo planner should be used instead (Schoemann, Boulton, & Short, 2017).

Value

A `data.frame` with rows for the recommended number of clusters `necessary_n_clusters`, the implied total sample size `total_N` (`necessary_n_clusters * cluster_size`), the target `width`, the intraclass correlation `icc`, the `cluster_size`, and the resulting `ci_width_expected` (the expected full CI width at the recommended size).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Raudenbush, S. W., & Liu, X.-F. (2001). Effects of study duration, frequency of observation, and sample size on power in studies of group differences in polynomial change. *Psychological Methods, 6*(4), 387–401. doi:10.1037/1082989X.6.4.387

Schoemann, A. M., Boulton, A. J., & Short, S. D. (2017). Determining power and sample size for simple and complex mediation models. *Social Psychological and Personality Science, 8*(4), 379–386. doi:10.1177/1948550617715068

Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel analysis: An introduction to basic and advanced multilevel modeling* (2nd ed.). Sage.

See Also

`ss_power_mixed_effects`, `var_icc`, `ss_aipe_icc`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: `power_fisher_exact()`, `ss_power_R2()`, `ss_power_R2_sensitivity()`, `ss_power_c()`, `ss_power_c_ancova()`, `ss_power_composite_ancova()`, `ss_power_composite_ancova_2group()`,

ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova
ss_power_composite_factorial_anova(), ss_power_composite_sem(), ss_power_contrast(),
ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(),
ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(),
ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(),
ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Other mixed models: R2_mixed_effects(), R2_mixed_effects_decomposition(), icc_lmer(),
manova_split_plot(), mixed_anova(), ss_aipe_mixed_effects_sensitivity(), ss_power_mixed_effects(),
ss_power_split_plot_anova()

Examples

```
# 1. Plan a two-level study with cluster size 20, ICC = 0.10,
#        sigma_y = 1, sigma_x = 1, target CI width = 0.20:
ss_aipe_mixed_effects(sigma2_y = 1, sigma2_x = 1, icc = 0.10,
                            width = 0.20, cluster_size = 20)

# 2. The same study with stronger clustering (ICC = 0.20):
ss_aipe_mixed_effects(sigma2_y = 1, sigma2_x = 1, icc = 0.20,
                            width = 0.20, cluster_size = 20)
```

---

ss_aipe_mixed_effects_sensitivity

Sensitivity Analysis for Sample Size Planning From the AIPE Perspective for a Mixed-Effects Fixed Effect

---

Description

Quantifies how much misspecification of the variance components (σ^2_Y, σ^2_X, and the intraclass correlation icc) distorts an AIPE-based sample size plan for a cluster-level fixed effect under a two-level random-intercept model. On each replication the function simulates K clusters of `cluster_size` observations each from

$$Y_{ki} = \beta\, X_k + u_k + \epsilon_{ki},$$

with $X_k \sim N(0, \sigma^2_X)$, $u_k \sim N(0, \text{icc} \cdot \sigma^2_Y)$, and $\epsilon_{ki} \sim N(0, (1 - \text{icc})\sigma^2_Y)$. The model is then refit by either `lme4::lmer` (if available) or by GLS-by-cluster aggregation, and a Wald CI on the fixed effect is recorded.

Usage

```
ss_aipe_mixed_effects_sensitivity(
  true_sigma2_y = NULL,
  true_sigma2_x = NULL,
  true_icc = NULL,
  true_beta = 0,
  estimated_sigma2_y = NULL,
  estimated_sigma2_x = NULL,
```

```
  estimated_icc = NULL,
  width,
  cluster_size = 20L,
  specified_K = NULL,
  conf_level = 0.95,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

`true_sigma2_y`	Population total variance of Y.
`true_sigma2_x`	Population variance of the cluster-level predictor.
`true_icc`	Population intraclass correlation (between-cluster share of total variance).
`true_beta`	Population fixed-effect slope (default `0`).
`estimated_sigma2_y`, `estimated_sigma2_x`, `estimated_icc`	Planning values passed to `ss_aipe_mixed_effects`; supply all three or `specified_K` but not both.
`width`	Desired full width of the CI on the fixed effect.
`cluster_size`	Number of observations per cluster (assumed balanced).
`specified_K`	Number of clusters to evaluate.
`conf_level`	Confidence level (default `0.95`).
`G`	Number of Monte Carlo replications.
`print_iter`	Logical.
`filename`	Optional path for a comma separated file recording every replication (the fixed effect estimate, the two confidence limits, the interval width, and two indicators of whether the interval missed `true_beta` below or above): nothing is written when `filename` is `NULL` (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at `tempfile(fileext = ".csv")`.

Value

A `data.frame` with rows for mean / median / SD of the realized fixed-effect estimate and CI width, the proportion of intervals at or below `width`, tail-specific and overall non-coverage of `true_beta`, and the input echoes.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

McNeish, D., & Kelley, K. (2019). Fixed effects versus mixed effects models for clustered data: Reviewing the approaches, disentangling the differences, and making recommendations. *Psychological Methods, 24*, 20–35. doi:10.1037/met0000182

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapters 15 and 16 on mixed-effects models.)

See Also

`ss_aipe_mixed_effects`, `icc_lmer`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: `ss_aipe_c_sensitivity()`, `ss_aipe_cliff_delta()`, `ss_aipe_cliff_delta_sensit`
`ss_aipe_composite_sem()`, `ss_aipe_equivalence_r()`, `ss_aipe_equivalence_r_sensitivity()`,
`ss_aipe_equivalence_smd()`, `ss_aipe_equivalence_smd_sensitivity()`, `ss_aipe_icc()`, `ss_aipe_icc_sensitivit`
`ss_aipe_indirect_effect()`, `ss_aipe_indirect_effect_sensitivity()`, `ss_aipe_omega_squared()`,
`ss_aipe_omega_squared_sensitivity()`, `ss_aipe_partial_r()`, `ss_aipe_partial_r_sensitivity()`,
`ss_aipe_pcm_sensitivity()`, `ss_aipe_r()`, `ss_aipe_r_sensitivity()`, `ss_aipe_reliability_sensitivity()`,
`ss_aipe_semipartial_r()`, `ss_aipe_semipartial_r_sensitivity()`

Other mixed models: `R2_mixed_effects()`, `R2_mixed_effects_decomposition()`, `icc_lmer()`,
`manova_split_plot()`, `mixed_anova()`, `ss_aipe_mixed_effects()`, `ss_power_mixed_effects()`,
`ss_power_split_plot_anova()`

Examples

```
# Monte Carlo sensitivity check, reduced sizes for a fast example.
set.seed(113)
ss_aipe_mixed_effects_sensitivity(
  true_sigma2_y = 1, true_sigma2_x = 1, true_icc = 0.10,
  true_beta = 0.30,
  specified_K = 25, cluster_size = 10,
  width = 0.40,
  G = 25, print_iter = FALSE
)
```

`ss_aipe_omega_squared` *Sample Size for AIPE on Omega Squared (ANOVA Effect Size)*

Description

Determines the sample size needed for the noncentral F confidence interval on the population omega squared (ω^2) to have a desired width (Accuracy in Parameter Estimation; Kelley, 2008; Steiger, 2004). The function uses the same noncentral F machinery as `ci_omega_squared`: at each candidate N it computes the expected CI width by inverting the noncentral F distribution and stops at the smallest N that achieves the target width.

Usage

```
ss_aipe_omega_squared(
  population_omega_squared,
  df_effect,
  width,
  which_width = c("Full", "Lower", "Upper"),
  conf_level = 0.95,
  assurance = NULL
)
```

Arguments

`population_omega_squared`	Anticipated population ω^2. Must lie in $[0, 1)$.
`df_effect`	Numerator degrees of freedom for the effect (e.g., $a - 1$ for an a-group one-way ANOVA, or the appropriate per-effect numerator df in a factorial design).
`width`	Desired full width of the CI on ω^2.
`which_width`	Whether `width` is the `"Full"` width of the interval (default) or a half-width: `"Lower"` and `"Upper"` both interpret `width` as half the full width, so they plan for a full width of twice `width` and return the same sample size. Because the interval is not generally symmetric about the estimate, its realized lower and upper half-widths can differ from each other and from half the full width; the planner does not target them separately. A genuinely one-sided width target is not currently offered.
`conf_level`	Desired confidence level (default `0.95`).
`assurance`	Optional. Probability that the realized CI is no wider than `width`; when supplied, the sample size is inflated by the standard chi squared correction (Kelley, 2008).

Details

Connection to noncentral *F* machinery. The CI on ω^2 is built by inverting the noncentral F sampling distribution of the observed F statistic, following Steiger (2004) and Kelley (2007); see `ci_omega_squared`. To plan a sample size, we iterate: for each candidate N, compute the F the analyst would observe *at the population effect size*, build its CI on ω^2, and stop at the smallest N whose CI width is below the target.

Population-effect-to-*F* mapping. Given a target ω^2, the expected sample F that yields exactly that ω^2 as the point estimate from $\hat{\omega}^2 = df_{\text{eff}}(F-1)/[df_{\text{eff}}(F-1)+N]$ is $F = 1+\omega^2 N/[df_{\text{eff}}(1-\omega^2)]$. This is the F value used at each iteration of the search.

Tolerance behavior at small *N*. For small candidate N the noncentral F lower limit is often clamped to zero (see `?ci_nc_F`). The search ignores these clamps in the iteration and reports the final clamp count, if any, as an informational message; this matches the convention in `ss_aipe_R2`.

Value

A `data.frame` with rows for the recommended *total* sample size `necessary_N`, the expected CI width at that sample size, and the inputs echoed back.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Algina, J., Moulder, B. C., & Moser, B. K. (2002). Sample size requirements for accurate estimation of squared semi-partial correlation coefficients. *Multivariate Behavioral Research, 37*(1), 37–57. doi:10.1207/s15327906mbr3701_02

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*(4), 524–555. doi:10.1080/00273170802490632

Kelley, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*, 137–152. doi:10.1037/a0028086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on η^2, Chapter 7 on factorial designs, and Chapter 11 on generalized η^2 for within-subjects designs.)

Steiger, J. H. (2004). Beyond the *F* test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

ci_omega_squared, omega_squared, omega_squared_partial, ss_aipe_R2

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(),
ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_icc_sensitivi
ss_aipe_indirect_effect(), ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(),
ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(), ss_aipe_partial_r_sensitivity(),
ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(), ss_aipe_reliability_sensitivity(),
ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# 1. Plan total N so the 95% CI on omega^2 has full width <= 0.10
#        in a 3-group one-way ANOVA (df_effect = 2), anticipated
#        omega^2 = 0.10.
ss_aipe_omega_squared(population_omega_squared = 0.10,
                      df_effect = 2,
                      width = 0.10)

# 2. Same problem with 80% assurance:
ss_aipe_omega_squared(population_omega_squared = 0.10,
                      df_effect = 2,
```

```
                    width = 0.10,
                    assurance = 0.80)
```

`ss_aipe_omega_squared_sensitivity` *Sensitivity Analysis for Sample Size Planning From the AIPE Perspective for Omega Squared*

Description

Quantifies how much misspecification of the population ω^2 distorts an AIPE-based sample size plan. The planner `ss_aipe_omega_squared` solves for the smallest N that yields an expected CI width below the target at the planning value. Here we generate `G` datasets from a balanced one-way ANOVA with population ω^2 =`true_omega_squared` and `df_effect + 1` groups at the planner-recommended N, compute the noncentral F confidence interval on each replication via `ci_omega_squared`, and summarize the realized widths and coverage of `true_omega_squared`.

Usage

```
ss_aipe_omega_squared_sensitivity(
  true_omega_squared = NULL,
  estimated_omega_squared = NULL,
  df_effect,
  width,
  specified_N = NULL,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

`true_omega_squared`	Population ω^2 (the data generating value); in $[0, 1)$.
`estimated_omega_squared`	ω^2 used to plan the study; supply this or `specified_N` but not both.
`df_effect`	Numerator degrees of freedom for the omnibus F, equal to the number of groups minus 1.
`width`	Desired full width of the confidence interval on ω^2.
`specified_N`	Total sample size to evaluate (incompatible with `estimated_omega_squared`).
`conf_level`	Confidence level (default `0.95`).
`assurance`	Optional assurance probability passed to `ss_aipe_omega_squared` when resolving the planned sample size.

G	Number of Monte Carlo replications (default 1000).
print_iter	Logical. Print iteration index per replication.
filename	Optional path for a comma separated file recording every replication (the sample $\hat{\omega}^2$, the two confidence limits, the interval width, and two indicators of whether the interval missed true_omega_squared below or above): nothing is written when filename is NULL (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at tempfile(fileext = ".csv").

Value

A data.frame with rows for mean / median / SD of the realized $\hat{\omega}^2$ and interval width, the proportion of intervals at or below width, tail-specific and overall empirical non-coverage of true_omega_squared, and the input echoes, including assurance (present only when an assurance was supplied).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on effect size measures.)

See Also

ss_aipe_omega_squared, ci_omega_squared

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(),
ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_icc_sensitivi
ss_aipe_indirect_effect(), ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(),
ss_aipe_omega_squared(), ss_aipe_partial_r(), ss_aipe_partial_r_sensitivity(), ss_aipe_pcm_sensitivity(
ss_aipe_r(), ss_aipe_r_sensitivity(), ss_aipe_reliability_sensitivity(), ss_aipe_semipartial_r(),
ss_aipe_semipartial_r_sensitivity()

Examples

```
# Well-specified: planner used omega^2 = 0.10, truth is 0.10.
# G is kept small here so the example runs quickly; raise it for a
# stable sensitivity estimate.
set.seed(113)
ss_aipe_omega_squared_sensitivity(
  true_omega_squared      = 0.10,
  estimated_omega_squared = 0.10,
  df_effect = 2, width = 0.10,
```

```
  G = 25, print_iter = FALSE
)
```

`ss_aipe_partial_r` *Sample Size for AIPE on a Partial Correlation*

Description

Determines the sample size needed for a confidence interval on a population partial correlation $\rho_{XY \cdot Z_1 \cdots Z_J}$ to have a desired width (Accuracy in Parameter Estimation; Kelley, 2008). The function inverts the asymptotic variance of the partial Pearson correlation, either on the raw scale (Olkin & Finn, 1995) or on the Fisher's *Z*-transformed scale (Fisher, 1921, 1924; Bonett, 2008), and solves for the smallest n that achieves the target half-width (or full width).

Usage

```
ss_aipe_partial_r(
  rho,
  J,
  width,
  which_width = c("Full", "Lower", "Upper"),
  conf_level = 0.95,
  fisher_z = FALSE,
  assurance = NULL
)
```

Arguments

`rho`	Anticipated population partial correlation, in $(-1, 1)$.
`J`	Number of variables partialled out (count of $Z_1, \ldots, Z_J$); must be at least 1.
`width`	Desired full width of the confidence interval on the partial correlation.
`which_width`	Whether `width` is the `"Full"` width of the interval (default) or a half-width: `"Lower"` and `"Upper"` both interpret `width` as half the full width, so they plan for a full width of twice `width` and return the same sample size. Because the interval is not generally symmetric about the estimate (with `fisher_z = TRUE` in particular), its realized lower and upper half-widths can differ from each other and from half the full width; the planner does not target them separately. A genuinely one-sided width target is not currently offered.
`conf_level`	Desired confidence level (default `0.95`).
`fisher_z`	Logical. If `TRUE`, the half-width target is applied on the Fisher-z scale (variance $1/(n-J-3)$; Bonett, 2008) and the resulting CI is back-transformed via $\tanh$. If `FALSE` (default), uses the raw-scale Olkin-Finn (1995) asymptotic variance.
`assurance`	Optional. Probability that the realized CI is no wider than `width` $(1-\gamma)$. When supplied, the sample size is inflated using the standard chi squared correction (Kelley, 2008); when `NULL`, the assurance is fixed at 0.5.

Details

Raw-scale Olkin-Finn asymptotic variance. The half-width of a $100(1-\alpha)\%$ CI on the partial Pearson correlation is approximately

$$w_{1/2} \approx z_{1-\alpha/2} \cdot \sqrt{\frac{(1-\rho_{XY\cdot Z}^2)^2}{n-J-1}}.$$

Solving for n:

$$n = J + 1 + \left\lceil (z_{1-\alpha/2})^2 \cdot (1-\rho_{XY\cdot Z}^2)^2 / w_{1/2}^2 \right\rceil.$$

This is the planning analog of the half-width of `ci_r` applied to a partial correlation.

Fisher-z scale (recommended for small ρ, near boundary, or small $n-J$). Bonett (2008) advocates planning on the variance-stabilized Fisher-z scale and back-transforming the bounds. On the Fisher's Z scale, the asymptotic half-width is

$$w_{1/2}^{(z)} \approx z_{1-\alpha/2}/\sqrt{n-J-3}.$$

Solving for the n that achieves a given back-transformed $w_{1/2}$ is done by a 1-D search; this is generally the more accurate route when n is small or $|\rho|$ is large.

When to use partial vs. simple correlation planning. Use this function when the inferential target is the population correlation between X and Y *after* statistically controlling for $Z_1, \ldots, Z_J$. For the simple Pearson correlation, see `ss_aipe_r`.

Note on conservatism of the assurance plan. The empirical simulation study of the AIPE planner family finds that `ss_aipe_partial_r()` is tight (zero overshoot) at 80% assurance but modestly conservative at 99% assurance, with an empirical ideal sample size of about 5 to 10 subjects smaller than the recommended sample size. The mechanism is the usual one for AIPE assurance plans: the Olkin-Finn (1995) Wald- style upper bound on $\Pr(\widehat{W} > \omega)$ that the planner inverts is not tight at the recommended sample size, especially at the 99% level where the inversion has to push further into the upper tail of $\widehat{W}$. The recommended sample size is a sufficient sample size rather than the smallest possible sample size. `ss_aipe_partial_r_sensitivity` quantifies the overshoot for any one condition.

Value

A `data.frame` with the rows `necessary_N` (the recommended total sample size, rounded up), `expected_width` at that sample size, and the inputs echoed back.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Algina, J., & Olejnik, S. (2003). Sample size tables for correlation analysis with applications in partial correlation and multiple regression analysis. *Multivariate Behavioral Research, 38*(3), 309–323. doi:10.1207/s15327906mbr3803_02

Bonett, D. G. (2008). Confidence intervals for standardized linear contrasts of means. *Psychological Methods, 13*(2), 99–109. doi:10.1037/1082989X.13.2.99

Fisher, R. A. (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron, 1*, 3–32.

Fisher, R. A. (1924). The distribution of the partial correlation coefficient. *Metron, 3*, 329–332.

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*(4), 524–555. doi:10.1080/00273170802490632

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on the one-way ANOVA and Chapter 4 on contrasts.)

Olkin, I., & Finn, J. D. (1995). Correlations redux. *Psychological Bulletin, 118*(1), 155–164. doi:10.1037/00332909.118.1.155

See Also

var_partial_r, expected_partial_r, ss_aipe_semipartial_r, ss_aipe_r

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(),
ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_icc_sensitivi
ss_aipe_indirect_effect(), ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(),
ss_aipe_omega_squared(), ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r_sensitivity(),
ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(), ss_aipe_reliability_sensitivity(),
ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# 1. Plan n so the 95% CI on rho_XY.Z (J = 2 controls) has
#        full width <= 0.20, when the anticipated partial r is 0.30.
ss_aipe_partial_r(rho = 0.30, J = 2, width = 0.20)

# 2. Same problem on the Fisher's Z scale (Bonett 2008):
ss_aipe_partial_r(rho = 0.30, J = 2, width = 0.20, fisher_z = TRUE)

# 3. With 80% assurance (Kelley 2008):
ss_aipe_partial_r(rho = 0.30, J = 2, width = 0.20, assurance = 0.80)
```

`ss_aipe_partial_r_sensitivity` *Sensitivity Analysis for Sample Size Planning From the AIPE Perspective for a Partial Correlation*

Description

Quantifies how much misspecification of the population partial correlation distorts an AIPE-based sample size plan. The function constructs an $(J+1) \times (J+1)$ population covariance matrix whose implied partial correlation between Y and X_1 (controlling for $X_2, \ldots, X_J$) equals `true_rho`, then on each replication draws an *n*-row sample from the corresponding multivariate normal distribution and computes the sample partial correlation and its Fisher's Z CI.

Usage

```
ss_aipe_partial_r_sensitivity(
  true_rho = NULL,
  estimated_rho = NULL,
  J,
  width,
  specified_N = NULL,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

`true_rho`	Population partial correlation between *Y* and X_1 controlling for $X_2, \ldots, X_J$; must lie in $(-1, 1)$.
`estimated_rho`	Planning value of the partial correlation passed to `ss_aipe_partial_r`; supply this or `specified_N` but not both.
`J`	Total number of predictors (so the partial correlation is between *Y* and one of the *J* predictors, partialing out the other $J-1$). Must be at least 1.
`width`	Desired full width of the CI on the partial correlation.
`specified_N`	Sample size to evaluate (incompatible with `estimated_rho`).
`conf_level`	Confidence level (default `0.95`).
`assurance`	Optional assurance probability passed to `ss_aipe_partial_r`.
`G`	Number of Monte Carlo replications (default 1000).
`print_iter`	Logical. Print iteration index per replication.

`filename` Optional path for a comma separated file recording every replication (the sample partial correlation, the two confidence limits, the interval width, and two indicators of whether the interval missed `true_rho` below or above): nothing is written when `filename` is `NULL` (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at `tempfile(fileext = ".csv")`.

Value

A `data.frame` with rows for the realized partial correlation, the interval width, the proportion of intervals at or below `width`, tail-specific and overall non-coverage of `true_rho`, and the input echoes, including `assurance` (present only when an assurance was supplied).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

`ss_aipe_partial_r`, `ss_aipe_semipartial_r_sensitivity`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: `ss_aipe_c_sensitivity()`, `ss_aipe_cliff_delta()`, `ss_aipe_cliff_delta_sensit`
`ss_aipe_composite_sem()`, `ss_aipe_equivalence_r()`, `ss_aipe_equivalence_r_sensitivity()`,
`ss_aipe_equivalence_smd()`, `ss_aipe_equivalence_smd_sensitivity()`, `ss_aipe_icc()`, `ss_aipe_icc_sensitivi`
`ss_aipe_indirect_effect()`, `ss_aipe_indirect_effect_sensitivity()`, `ss_aipe_mixed_effects_sensitivity()`,
`ss_aipe_omega_squared()`, `ss_aipe_omega_squared_sensitivity()`, `ss_aipe_partial_r()`,
`ss_aipe_pcm_sensitivity()`, `ss_aipe_r()`, `ss_aipe_r_sensitivity()`, `ss_aipe_reliability_sensitivity()`,
`ss_aipe_semipartial_r()`, `ss_aipe_semipartial_r_sensitivity()`

Examples

```
# Reduced replications and a wide target interval keep this fast.
set.seed(113)
ss_aipe_partial_r_sensitivity(
  true_rho = 0.40, estimated_rho = 0.40, J = 3, width = 0.40,
  G = 50, print_iter = FALSE
)
```

ss_aipe_pcm *Sample Size Planning for Polynomial Change Models in Longitudinal Study*

Description

This function plans sample size with respect to the group-by-time interaction in the context of a longitudinal design with two groups. It plans sample size from the accuracy in parameter estimation (AIPE) perspective, where the goal is to obtain a sufficiently narrow confidence interval for the fixed effect polynomial change coefficient parameter (e.g., linear, quadratic, etc.). The sample size returned can be one such that (a) the expected confidence interval width is sufficiently narrow, or (b) the observed confidence interval will be sufficiently narrow with a specified high degree of assurance (e.g., .99, .95, .90, etc.). This function accompanies Kelley and Rausch (2011).

Usage

```
ss_aipe_pcm(
  variance_trend,
  error_variance = NULL,
  variance_true_minus_estimated_trend = NULL,
  duration,
  frequency,
  width,
  conf_level = 0.95,
  trend = "linear",
  assurance = NULL
)
```

Arguments

variance_trend	The variance of the individuals' true change coefficients (i.e., $\sigma^2_{\upsilon_m}$ in Kelley & Rausch, 2011) for the polynomial trend (e.g., linear, quadratic, etc.) of interest
error_variance	The true level one error variance (i.e., σ^2_ϵ in Kelley & Rausch, 2011). Either error_variance or variance_true_minus_estimated_trend must be supplied; if variance_true_minus_estimated_trend is given directly, error_variance may be omitted.
variance_true_minus_estimated_trend	The variance of the difference between the mth true change coefficient minus the mth estimated change coefficient (i.e., $\sigma^2_{\tilde{\pi}_m - \pi_m}$ from Equation 19 in Kelley & Rausch, 2011). When derived from error_variance this equals $\sigma^2_\epsilon f^{2p} / \sum_t c^2_{mt}$, where f is the frequency, p the polynomial order, and $\sum_t c^2_{mt}$ the sum of squared orthogonal polynomial contrast weights over the measurement occasions. A user who already has this variance may supply it directly and omit error_variance.
duration	The duration of the study
frequency	The number of times measurement occurs within each unit of time

`width`	Width of the confidence interval
`conf_level`	The desired level of confidence for the confidence interval that will be computed at the completion of the study
`trend`	The polynomial trend (1st-3rd) of interest specified as "linear", "quadratic", or "cubic"
`assurance`	Value with which confidence can be placed that describes the likelihood of obtaining a confidence interval less than the value specified (e.g, .80, .90, .95)

Value

A `data.frame` (class `dmar_tbl`) with a single row, `necessary_n_per_group`, giving the necessary number of subjects *per group* (the total study size is twice this value) for the combination of the desired confidence interval width, confidence level, optional assurance, and the population parameters at the specified design.

Note

Like in all formal sample size planning methods that require the value of one or more population parameter(s), if the population parameters are incorrectly specified, there is no guarantee that the sample size this function returns will be accurate. Of course, the further away from the true values, the further away the true sample size will tend to be.

The number of timepoints in a study (say M) is defined by $f \times D + 1$, where f is the frequency and D is the duration.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Kelley, K., & Rausch, J. R. (2011). Sample size planning for longitudinal models: Accuracy in parameter estimation for polynomial change parameters. *Psychological Methods, 16*(4), 391–405. doi:10.1037/a0023352

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapters 11, 15.)

Raudenbush, S. W., & Liu, X.-F. (2001). Effects of study duration, frequency of observation, and sample size on power in studies of group differences in polynomial change. *Psychological Methods, 6*(4), 387–401. doi:10.1037/1082989X.6.4.387

See Also

`ss_power_pcm` for the power analytic analog (planning to detect the group-by-time change difference rather than to estimate it precisely) on the same model, and `ss_aipe_pcm_sensitivity` for a Monte Carlo check of how parameter misspecification affects the plan.

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# The examples reproduce the tolerance-of-antisocial-thinking illustration
# of Kelley and Rausch (2011, Tables 1 and 2), which draws on the National
# Youth Survey data also used by Raudenbush and Liu (2001). The level-one
# error variance is 0.0262 and the between-subject slope variance is 0.003.
# The planner finds the sample size needed for a confidence interval on the
# group-by-time slope difference that is no wider than `width`. The returned
# necessary_n_per_group is per group, so the total study size is twice that. Unlike
# power analysis, the value of the slope is not needed here: the confidence
# interval width does not depend on it.

# (1) Expected-width planning. With five measurement occasions
#     (M = frequency * duration + 1 = 1 * 4 + 1) and a target width of
#     0.025, the expected 95% confidence interval is sufficiently narrow at
#     278 subjects per group (Kelley & Rausch, 2011, Table 1, T = 5).
ss_aipe_pcm(variance_trend = 0.003, error_variance = 0.0262,
            duration = 4, frequency = 1, width = 0.025, conf_level = .95)

# (2) More measurement occasions sharpen the estimate. Extending the study
#     so that M = 10 (duration = 9, frequency = 1) cuts the expected-width
#     requirement from 278 to 165 per group (Kelley & Rausch, 2011, Table 1,
#     T = 10).
ss_aipe_pcm(variance_trend = 0.003, error_variance = 0.0262,
            duration = 9, frequency = 1, width = 0.025, conf_level = .95)

# (3) A wider tolerated interval costs less. Relaxing the target width from
#     0.025 to 0.05 at M = 5 drops the requirement from 278 to 71 per group
#     (Kelley & Rausch, 2011, Table 1, T = 5).
ss_aipe_pcm(variance_trend = 0.003, error_variance = 0.0262,
            duration = 4, frequency = 1, width = 0.05, conf_level = .95)

# (4) Adding an assurance parameter. Requiring 85% assurance that the
#     realized confidence interval will be no wider than 0.025 raises the
#     M = 5 requirement from 278 to 295 per group (Kelley & Rausch, 2011,
#     Table 2, T = 5). Assurance guards against the expected-width plan being
#     too small for the particular sample obtained.
ss_aipe_pcm(variance_trend = 0.003, error_variance = 0.0262,
            duration = 4, frequency = 1, width = 0.025, conf_level = .95,
            assurance = .85)

# (5) A higher assurance costs more. Demanding 99% assurance rather than 85%
#     raises the per-group requirement further, from 295 to 316.
ss_aipe_pcm(variance_trend = 0.003, error_variance = 0.0262,
            duration = 4, frequency = 1, width = 0.025, conf_level = .95,
            assurance = .99)
```

ss_aipe_pcm_sensitivity

Sensitivity Analysis for Sample Size Planning From the AIPE Perspective for a Polynomial Change Parameter

Description

Quantifies how much misspecification of the population between-subject slope variance and within-subject error variance distorts an AIPE-based sample size plan for the group-by-time polynomial change parameter. On each replication the function simulates two independent groups of n subjects each, measured at $M = f \times D + 1$ timepoints, where every subject has a true linear slope drawn from $N(0, \text{true\_variance\_trend})$ and within-subject observations have residual variance `true_error_variance`. Subject-level OLS slopes are computed in each group, the between-group difference in mean slopes (the change parameter β_{m1} that `ss_aipe_pcm` plans for) is estimated, and a two-group t-confidence interval on that difference (pooled standard error, $2n - 2$ degrees of freedom) is recorded. The function only handles `trend = "linear"` in the simulator; for quadratic / cubic trends, the planner's closed-form solution is still available via `ss_aipe_pcm`.

Usage

```
ss_aipe_pcm_sensitivity(
  true_variance_trend = NULL,
  true_error_variance = NULL,
  estimated_variance_trend = NULL,
  estimated_error_variance = NULL,
  duration,
  frequency,
  width,
  n_per_group = NULL,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

`true_variance_trend`
: Population between-subject variance of the polynomial change coefficient (the data generating $\sigma^2_{v_m}$ of Kelley & Rausch, 2011).

`true_error_variance`
: Population within-subject error variance (σ^2_ϵ).

`estimated_variance_trend`
: Planning value of `variance_trend` passed to `ss_aipe_pcm`; supply this and `estimated_error_variance`, or supply `n_per_group`.

`estimated_error_variance`
: Planning value of `error_variance` passed to `ss_aipe_pcm`.

`duration`
: Study duration (in time units).

`frequency`
: Number of measurements per unit time. Total timepoints = $f \times D + 1$.

`width`
: Desired full width of the CI on the between-group difference in change parameters (β_{m1}).

`n_per_group`
: Number of subjects to evaluate (incompatible with the estimated-variance arguments).

`conf_level`
: Confidence level (default `0.95`).

`assurance`
: Optional assurance probability passed to the planner.

`G`
: Number of Monte Carlo replications.

`print_iter`
: Logical.

`filename`
: Optional path to a CSV file; when supplied, the per-replication results (the slope difference, its interval limits and width, and the two tail misses) are written there, appended when the file already exists, and a throwaway run should point it at `tempfile(fileext = ".csv")`; the default `NULL` writes nothing.

Value

A `data.frame` with rows for mean / median / SD of the realized estimated slope difference and CI width, the proportion of intervals at or below `width`, tail-specific and overall non-coverage of the population slope difference (0 by construction in this simulator), and the input echoes, including `assurance` (present only when an assurance was supplied).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Rausch, J. R. (2011). Sample size planning for longitudinal models: Accuracy in parameter estimation for polynomial change parameters. *Psychological Methods, 16*(4), 391–405. doi:10.1037/a0023352

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapters 11 and 15.)

See Also

`ss_aipe_pcm`, `ss_aipe_mixed_effects_sensitivity`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: `ss_aipe_c_sensitivity()`, `ss_aipe_cliff_delta()`, `ss_aipe_cliff_delta_sensit`
`ss_aipe_composite_sem()`, `ss_aipe_equivalence_r()`, `ss_aipe_equivalence_r_sensitivity()`,
`ss_aipe_equivalence_smd()`, `ss_aipe_equivalence_smd_sensitivity()`, `ss_aipe_icc()`, `ss_aipe_icc_sensitivi`
`ss_aipe_indirect_effect()`, `ss_aipe_indirect_effect_sensitivity()`, `ss_aipe_mixed_effects_sensitivity()`,
`ss_aipe_omega_squared()`, `ss_aipe_omega_squared_sensitivity()`, `ss_aipe_partial_r()`,

ss_aipe_partial_r_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(), ss_aipe_reliability_sensitivity(
ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# Every replication simulates two full groups of subjects, fits a slope
# for each subject, and forms a confidence interval on the difference in
# mean slopes. G = 20 keeps the example quick; a reported sensitivity
# study deserves the default G = 1000. With the planning values equal to
# the population values, the realized mean width should sit at or just
# under the target.
set.seed(113)
ss_aipe_pcm_sensitivity(
  true_variance_trend = 0.003, true_error_variance = 0.0262,
  estimated_variance_trend = 0.003, estimated_error_variance = 0.0262,
  duration = 4, frequency = 1, width = 0.05,
  G = 20, print_iter = FALSE
)
```

---

ss_aipe_r *Sample Size for AIPE on a Pearson Correlation*

Description

Determines the sample size needed for a confidence interval on a population Pearson correlation ρ to have a desired width (accuracy in parameter estimation; Kelley & Maxwell, 2003). The interval planned for is the Fisher's *Z* interval that correlations_test reports for method = "pearson": the correlation is transformed as $z(r) = \operatorname{atanh}(r)$, an interval with standard error $1/\sqrt{n-3}$ is formed on the z scale, and the limits are back-transformed through $\tanh(\cdot)$ (Fisher, 1921; Bonett & Wright, 2000, Equation 2). Because the plan targets the same interval the analysis will report, the planned width and the analyzed width agree.

Usage

```
ss_aipe_r(rho, width, conf_level = 0.95, assurance = NULL)
```

Arguments

rho	Anticipated population Pearson correlation, in $(-1, 1)$.
width	Desired full width of the confidence interval on the correlation.
conf_level	Desired confidence level (default 0.95).
assurance	Optional. Probability that the realized CI is no wider than width $(1-\gamma)$. When supplied, the sample size is inflated using the standard chi squared correction (Kelley, 2008); when NULL, the assurance is fixed at 0.5.

Details

Closed-form first pass. On the Fisher's Z scale the interval has half-width $z_{1-\alpha/2}/\sqrt{n-3}$, and the delta method maps it back to the correlation scale as approximately

$$w \approx 2\, z_{1-\alpha/2}\, \frac{1-\rho^2}{\sqrt{n-3}},$$

which solves to the first-stage approximation of Bonett and Wright (2000),

$$n_0 = 3 + \left\lceil 4\,(z_{1-\alpha/2})^2 (1-\rho^2)^2 / w^2 \right\rceil.$$

Exact iteration. The back-transformed width depends on ρ through $\tanh(\cdot)$, so the delta method approximation can land a few observations off in either direction. Starting from n_0, the function evaluates the exact back-transformed width $\tanh(z_\rho + z_{1-\alpha/2}/\sqrt{n-3}) - \tanh(z_\rho - z_{1-\alpha/2}/\sqrt{n-3})$ and steps the integer n until it is the smallest sample size whose width is at or below `width`. Where Bonett and Wright (2000) stop after a single second-stage adjustment, this search is exact.

The planning value matters least near zero. At a fixed sample size the back-transformed width is largest at $\rho = 0$ and shrinks as $|\rho|$ grows, so a planning value closer to zero yields a larger, more conservative sample size. When little is known about the population correlation, `rho = 0` gives the sample size that suffices for any population value.

When to use simple vs. partial correlation planning. Use this function when the inferential target is the correlation between two variables with nothing partialed out. When the target is the correlation after statistically controlling for other variables, see `ss_aipe_partial_r`.

The Monte Carlo companion `ss_aipe_r_sensitivity` evaluates how the plan behaves when the population correlation differs from the planning value.

Value

A `data.frame` with the rows `necessary_N` (the recommended total sample size, rounded up), `expected_width` at that sample size, and the inputs echoed back.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bonett, D. G., & Wright, T. A. (2000). Sample size requirements for estimating Pearson, Kendall and Spearman correlations. *Psychometrika, 65*(1), 23–28. doi:10.1007/BF02294183

Fisher, R. A. (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron, 1*, 3–32.

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*(4), 524–555. doi:10.1080/00273170802490632

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9 on correlations.)

See Also

ss_aipe_r_sensitivity, correlations_test, ss_aipe_partial_r, ss_power_r, convert_r_Z

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(),
ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_icc_sensitivi
ss_aipe_indirect_effect(), ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(),
ss_aipe_omega_squared(), ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(),
ss_aipe_partial_r_sensitivity(), ss_aipe_pcm_sensitivity(), ss_aipe_r_sensitivity(),
ss_aipe_reliability_sensitivity(), ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# Plan n so the 95% CI on the Pearson correlation has full width
# at most 0.20, when the anticipated correlation is 0.30.
ss_aipe_r(rho = 0.30, width = 0.20)

# A narrower target width requires a larger sample size.
ss_aipe_r(rho = 0.30, width = 0.10)

# With 80% assurance that the realized interval is no wider than
# the target (Kelley, 2008):
ss_aipe_r(rho = 0.30, width = 0.20, assurance = 0.80)

# Planning at rho = 0 gives the sample size that suffices for any
# population correlation, since the interval is widest there.
ss_aipe_r(rho = 0, width = 0.20)
```

---

ss_aipe_r_sensitivity *Sensitivity Analysis for Sample Size Planning From the AIPE Perspective for a Pearson Correlation*

Description

Quantifies how much misspecification of the population Pearson correlation distorts an AIPE-based sample size plan. On each replication the function draws an n-row sample from a bivariate normal distribution with correlation `true_rho` and computes the sample correlation and its Fisher's Z CI, the interval ss_aipe_r plans for and correlations_test reports. Because the back-transformed width is largest at $\rho = 0$ and shrinks as $|\rho|$ grows, a planning value whose magnitude overstates the population correlation yields realized intervals wider than planned, and the summary rows report by how much.

Usage

```
ss_aipe_r_sensitivity(
  true_rho = NULL,
  estimated_rho = NULL,
  width,
  specified_N = NULL,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

`true_rho`	Population Pearson correlation; must lie in $(-1, 1)$.
`estimated_rho`	Planning value of the correlation passed to `ss_aipe_r`; supply this or `specified_N` but not both.
`width`	Desired full width of the CI on the correlation.
`specified_N`	Sample size to evaluate (incompatible with `estimated_rho`).
`conf_level`	Confidence level (default `0.95`).
`assurance`	Optional assurance probability passed to `ss_aipe_r`.
`G`	Number of Monte Carlo replications (default 1000).
`print_iter`	Logical. Print iteration index per replication.
`filename`	Optional path for a comma separated file recording every replication (the sample correlation, the two confidence limits, the interval width, and two indicators of whether the interval missed `true_rho` below or above): nothing is written when `filename` is `NULL` (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at `tempfile(fileext = ".csv")`.

Value

A `data.frame` with rows for the realized correlation, the interval width, the proportion of intervals at or below `width`, tail-specific and overall non-coverage of `true_rho`, and the input echoes, including `assurance` (present only when an assurance was supplied).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bonett, D. G., & Wright, T. A. (2000). Sample size requirements for estimating Pearson, Kendall and Spearman correlations. *Psychometrika, 65*(1), 23–28. doi:10.1007/BF02294183

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

ss_aipe_r, ss_aipe_partial_r_sensitivity

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(),
ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_icc_sensitivi
ss_aipe_indirect_effect(), ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(),
ss_aipe_omega_squared(), ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(),
ss_aipe_partial_r_sensitivity(), ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_reliability_sensitivi
ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# Reduced replications and a wide target interval keep this fast.
set.seed(113)
ss_aipe_r_sensitivity(
  true_rho = 0.30, estimated_rho = 0.30, width = 0.40,
  G = 50, print_iter = FALSE
)
```

---

ss_aipe_R2 *Sample Size Planning for Accuracy in Parameter Estimation for the Multiple Correlation Coefficient*

Description

Determines necessary sample size for the multiple correlation coefficient so that the confidence interval for the population multiple correlation coefficient is sufficiently narrow. Optionally, there is a certainty parameter that allows one to be a specified percent certain that the observed interval will be no wider than desired.

Usage

```
ss_aipe_R2(
  population_R2 = NULL,
  conf_level = 0.95,
  width = NULL,
  random_predictors = TRUE,
```

```
  which_width = "Full",
  p = NULL,
  assurance = NULL,
  verify_ss = FALSE,
  tol = 1e-09,
  ...
)
```

Arguments

population_R2	Value of the population multiple correlation coefficient
conf_level	Confidence interval level (e.g., .95, .99, .90); 1-Type I error rate
width	Width of the confidence interval (see which_width)
random_predictors	Whether or not the predictor variables are random (set to TRUE) or are fixed (set to FALSE)
which_width	Defines the width that width refers to
p	The number of predictor variables
assurance	Value with which confidence can be placed that describes the likelihood of obtaining a confidence interval less than the value specified (e.g, .80, .90, .95)
verify_ss	Evaluates numerically via an internal Monte Carlo simulation the exact sample size given the specifications
tol	The tolerance of the iterative function ci_nc_t for convergence
...	For modifying the parameters of functions this function calls upon

Details

This function determines a necessary sample size so that the expected confidence interval width for the squared multiple correlation coefficient is sufficiently narrow (when assurance=NULL) so that the obtained confidence interval is no larger than the value specified with some desired degree of certainty (i.e., a probability that the obtained width is less than the specified width). The method depends on whether or not the regressors are regarded as fixed or random. This is the case because the distribution theory for the two cases is different and thus the confidence interval procedure is conditional on the type of regressors. The default methods are approximate but can be made exact with the specification of verify_ss=TRUE, which performs an a priori Monte Carlo simulation study. Kelley (2008) and Kelley & Maxwell (2008) detail the methods used in the function, with the former focusing on random regressors and the latter on fixed regressors.

It is recommended that the option verify_ss should always be used! Doing so uses the method implied sample size as an estimate and then evaluates with an internal Monte Carlo simulation (i.e., via "brute-force" methods) the exact sample size given the goals specified. When verify_ss=TRUE, the default number of iterations is 10,000 but this can be changed by specifying G=5000 (or some other value; 10000 is the recommended). When verify_ss=TRUE is specified, an internal function verify_ss_aipe_r2 calls upon the ss_aipe_R2_sensitivity function for purposes of the internal Monte Carlo simulation study. Two of its arguments pass through ...: g (default 500), the number of replications used for each candidate N in the coarse search that brackets the answer, and G (default 10000), the number used in the final pass that confirms the sample size near that bracket.

Value

A 1-row `data.frame` with columns `term` and `value`. The `term` value is `"necessary_N"` and `value` is the necessary total sample size N given the input specifications.

Note

With `verify_ss = TRUE` the function can take some time to converge (e.g., several minutes to a quarter hour) because the closed form approximation is followed by an a priori Monte Carlo simulation. The default `verify_ss = FALSE` returns the closed form approximation only and is essentially instantaneous.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Algina, J. & Olejnik, S. (2000). Determining sample size for accurate estimation of the squared multiple correlation coefficient. *Multivariate Behavioral Research, 35*, 119–137. doi:10.1207/s15327906mbr3501_5

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*, 524–555. doi:10.1080/00273170802490632

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on R^2 as a model comparison effect size.)

Steiger, J. H., & Fouladi, R. T. (1992). R2: A computer program for interval estimation, power calculations, sample size estimation, and hypothesis testing in multiple regression. *Behavior Research Methods, Instruments, & Computers, 24*(4), 581–582. doi:10.3758/BF03203611

See Also

`ci_R2`, `ci_nc_t`, `ss_aipe_R2_sensitivity`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# 1. Closed form planner under random predictors (the typical case).
#    Sample size sufficient for the expected CI width on rho^2 to be .10.
ss_aipe_R2(population_R2 = .50, conf_level = .95, width = .10,
           which_width = "Full", p = 5, random_predictors = TRUE)
```

```
# 2. The same target under fixed predictors (planned dosing levels,
#    factorial covariates, and the like) needs a smaller N, since fixed
#    predictors contribute no sampling variability of their own.
ss_aipe_R2(population_R2 = .50, conf_level = .95, width = .10,
           which_width = "Full", p = 5, random_predictors = FALSE)

# 3. An assurance of .85, so that the realized width is no larger than the
#    target in 85 percent of replications rather than only on average,
#    needs a larger N than the expected width plan in (1).
ss_aipe_R2(population_R2 = .50, conf_level = .95, width = .10,
           which_width = "Full", p = 5, assurance = .85,
           random_predictors = TRUE)

# 4. verify_ss = TRUE follows the closed form approximation with an a
#    priori Monte Carlo simulation of the realized width, starting from
#    the closed form answer and returning the sample size the simulation
#    settles on, which is what a plan meant to be defended deserves. The
#    coarse search runs g replications per candidate N and the final pass
#    runs G; the small counts here keep the example quick, and a reported
#    plan deserves the defaults of g = 500 and G = 10000.
set.seed(113)
ss_aipe_R2(population_R2 = .50, conf_level = .95, width = .10,
           which_width = "Full", p = 5, random_predictors = TRUE,
           verify_ss = TRUE, g = 10, G = 30)
```

`ss_aipe_R2_sensitivity` *Sensitivity Analysis for Sample Size Planning With the Goal of Accuracy in Parameter Estimation (I.e., a Narrow Observed Confidence Interval)*

Description

Given `estimated_R2` and `true_R2`, one can perform a sensitivity analysis to determine the effect of a misspecified population squared multiple correlation coefficient using the Accuracy in Parameter Estimation (AIPE) approach to sample size planning. The function evaluates the effect of a misspecified `true_R2` on the width of obtained confidence intervals.

Usage

```
ss_aipe_R2_sensitivity(
  true_R2 = NULL,
  estimated_R2 = NULL,
  w = NULL,
  p = NULL,
  random_predictors = TRUE,
  specified_N = NULL,
```

```
  assurance = NULL,
  conf_level = 0.95,
  generate_random_predictors = TRUE,
  rho_yx = 0.3,
  rho_xx = 0.3,
  G = 10000,
  print_iter = TRUE,
  filename = NULL
)
```

Arguments

true_R2	Value of the population squared multiple correlation coefficient
estimated_R2	Value of the estimated (for sample size planning) squared multiple correlation coefficient
w	Full confidence interval width of interest
p	Number of predictors
random_predictors	Whether or not the sample size procedure and the simulation itself should be based on random (set to TRUE) or fixed predictors (set to FALSE)
specified_N	Selected sample size to use in order to determine distributional properties at a given value of sample size
assurance	Parameter to ensure confidence interval width with a specified degree of certainty
conf_level	Confidence interval coverage (symmetric coverage)
generate_random_predictors	Specify whether the simulation should be based on random (default) or fixed regressors.
rho_yx	Value of the correlation between *y* (dependent variable) and each of the *x* variables (independent variables)
rho_xx	Value of the correlation among the *x* variables (independent variables)
G	Number of generations (i.e., replications) of the simulation
print_iter	Should the iteration number (between 1 and G) during the run of the function
filename	Optional path of a CSV file to receive the per-replication results (the confidence limits, the observed R^2, and the one-sided and full interval widths), overwriting any file already at that path; the default NULL writes nothing, and a throwaway run that wants the file should point it at tempfile(fileext = ".csv").

Details

When estimated_R2=true_R2, the results are that of a simulation study when all assumptions are satisfied. Rather than specifying estimated_R2, one can specify specified_N to determine the results of a particular sample size (when doing this estimated_R2 cannot be specified).

The sample size estimation procedure technically assumes multivariate normal variables (p+1) with fixed predictors (x/independent variables), yet the function assumes random multivariate normal

predictors (having a p+1 multivariate distribution). As Gatsonis and Sampson (1989) note in the context of statistical power analysis (recall this function is used in the context of precision), there is little difference in the outcome.

In the behavioral, educational, and social sciences, predictor variables are almost always random, and thus random_predictors should generally be used. random_predictors=TRUE specifies how both the sample size planning procedure and the confidence intervals are calculated based on the random predictors/regressors. The internal simulation generates random or fixed predictors/regressors based on whether variables predictor variables are random or fixed. However, when random_predictors=FALSE, only the sample size planning procedure and the confidence intervals are calculated based on the parameter. The parameter generate_random_predictors (where the default is TRUE so that random predictors/regressors are generated) allows random or fixed predictor variables to be generated. Because the sample size planning procedure and the internal simulation are both specified, for purposes of sensitivity analysis random/fixed can be crossed to examine the effects of specifying sample size based on one but using it on data based on the other.

Value

A data.frame with columns term and value summarizing the Monte Carlo sensitivity analysis across G replications. The term entries are: mean_lower_limit, median_lower_limit, sd_lower_limit, mean_upper_limit, median_upper_limit, sd_upper_limit (summaries of the realized confidence limits); mean_R2, median_R2, sd_R2 (summaries of the observed R^2); mean_ci_width_lower, median_ci_width_lower, sd_ci_width_lower, mean_ci_width_upper, median_ci_width_upper, sd_ci_width_upper (summaries of the one-sided widths, measured from the observed R^2 to each limit); mean_ci_width, median_ci_width, sd_ci_width (summaries of the full interval widths); pct_ci_less_w (proportion of intervals with width at or below the planning target w); pct_ci_miss_low and pct_ci_miss_high (tail-specific empirical non-coverage of true_R2); total_type_I_error (overall empirical non-coverage, the sum of the two tails); num_probs_with_cis (number of replications on which a confidence interval could not be obtained); and the input echoes total_N (the sample size evaluated), p, true_R2, estimated_R2 (NA when specified_N was supplied instead), width, conf_level, and assurance (present only when an assurance was supplied). The proportion rows are on the 0 to 1 scale, not percentages.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Algina, J. & Olejnik, S. (2000). Determining sample size for accurate estimation of the squared multiple correlation coefficient. *Multivariate Behavioral Research, 35*, 119–137. doi:10.1207/s15327906mbr3501_5

Gatsonis, C. & Sampson, A. R. (1989). Multiple Correlation: Exact power and sample size calculations. *Psychological Bulletin, 106*(3), 516–524.

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals, *Multivariate Behavioral Research, 43*(4), 524–555. doi:10.1080/00273170802490632

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on R^2 as a model comparison effect size.)

Steiger, J. H., & Fouladi, R. T. (1992). R2: A computer program for interval estimation, power calculations, sample size estimation, and hypothesis testing in multiple regression. *Behavior Research Methods, Instruments, & Computers, 24*(4), 581–582. doi:10.3758/BF03203611

See Also

ci_R2, ci_nc_t, ss_aipe_R2

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# The planner used .4 for a population squared multiple correlation
# coefficient that is really .5. G = 25 keeps the example quick; a
# reported sensitivity analysis deserves the default G of 10000.
set.seed(113)
ss_aipe_R2_sensitivity(true_R2 = .5, estimated_R2 = .4, w = .10, p = 5,
                       conf_level = 0.95, G = 25, print_iter = FALSE)
```

---

ss_aipe_rc *Sample Size Necessary for the Accuracy in Parameter Estimation Approach for an Unstandardized Regression Coefficient of Interest*

Description

A function used to plan sample size from the accuracy in parameter estimation perspective for an unstandardized regression coefficient of interest given the input specification.

Usage

```
ss_aipe_rc(
  rho2_Y_X = NULL,
  Rho2_j_X_without_j = NULL,
  p = NULL,
  b_j = NULL,
  width,
  which_width = "Full",
  sigma_Y = 1,
  sigma_X_j = 1,
  rho_XX = NULL,
```

```
  rho_YX = NULL,
  which_predictor = NULL,
  alpha_lower = NULL,
  alpha_upper = NULL,
  conf_level = 0.95,
  assurance = NULL
)
```

Arguments

rho2_Y_X	Population value of the squared multiple correlation coefficient
Rho2_j_X_without_j	Population value of the squared multiple correlation coefficient predicting the *j*th predictor variable from the remaining *p*-1 predictor variables
p	The number of predictor variables
b_j	The regression coefficient for the *j*th predictor variable (i.e., the predictor of interest)
width	The desired width of the confidence interval
which_width	Which Width ("Full", "Lower", or "Upper") the width refers to (at present, only "Full" can be specified)
sigma_Y	The population standard deviation of *Y* (i.e., the dependent variables)
sigma_X_j	The population standard deviation of the *j*th *X* variable (i.e., the predictor variable of interest)
rho_XX	Population correlation matrix for the *p* predictor variables
rho_YX	Population *p* length vector of correlation between the dependent variable (*Y*) and the *p* independent variables
which_predictor	Identifies which of the *p* predictors is of interest
alpha_lower	Type I error rate for the lower confidence interval limit
alpha_upper	Type I error rate for the upper confidence interval limit
conf_level	Desired level of confidence for the computed interval (i.e., 1 - the Type I error rate)
assurance	Degree of certainty that the obtained confidence interval will be sufficiently narrow

Details

Not all of the arguments need to be specified, only those that provide all of the necessary information so that the sample size can be determined for the conditions specified.

Value

Returns the necessary sample size in order for the goals of accuracy in parameter estimation to be satisfied for the confidence interval for a particular regression coefficient given the input specifications.

Note

This function calls upon ss_aipe_reg_coef in DMAR but has a different naming scheme. See ss_aipe_reg_coef for more details.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means and Chapter 6 on trend analysis.)

See Also

ss_aipe_reg_coef_sensitivity, ci_nc_t, ss_aipe_reg_coef, ss_aipe_src

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Exchangeable correlation structure
rho_YX <- c(.3, .3, .3, .3, .3)
rho_XX <- rbind(c(1, .5, .5, .5, .5), c(.5, 1, .5, .5, .5), c(.5, .5, 1, .5, .5),
c(.5, .5, .5, 1, .5), c(.5, .5, .5, .5, 1))

ss_aipe_rc(width = .1, which_width = "Full", sigma_Y = 1, sigma_X = 1, rho_XX = rho_XX,
           rho_YX = rho_YX, which_predictor = 1, conf_level = 1 - .05)

ss_aipe_rc(width = .1, which_width = "Full", sigma_Y = 1, sigma_X = 1, rho_XX = rho_XX,
           rho_YX = rho_YX, which_predictor = 1, conf_level = 1 - .05, assurance = .85)
```

---

ss_aipe_rc_sensitivity

Sensitivity Analysis for Sample Size Planning From the Accuracy in Parameter Estimation Perspective for the Unstandardized Regression Coefficient

---

Description

Performs a sensitivity analysis when planning sample size from the Accuracy in Parameter Estimation Perspective for the unstandardized regression coefficient.

Usage

```
ss_aipe_rc_sensitivity(
  true_var_Y = NULL,
  true_cov_YX = NULL,
  true_cov_XX = NULL,
  estimated_var_Y = NULL,
  estimated_cov_YX = NULL,
  estimated_cov_XX = NULL,
  specified_N = NULL,
  which_predictor = 1,
  w = NULL,
  noncentral = FALSE,
  standardize = FALSE,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
  print_iter = TRUE,
  filename = NULL
)
```

Arguments

true_var_Y	Population variance of the dependent variable (Y)
true_cov_YX	Population covariances vector between the p predictor variables and the dependent variable (Y)
true_cov_XX	Population covariance matrix of the p predictor variables
estimated_var_Y	Estimated variance of the dependent variable (Y)
estimated_cov_YX	Estimated covariances vector between the p predictor variables and the dependent variable (Y)
estimated_cov_XX	Estimated Population covariance matrix of the p predictor variables
specified_N	Directly specified sample size (instead of planning one from the estimated covariance structure)
which_predictor	identifies which of the p predictors is of interest
w	desired confidence interval width for the regression coefficient of interest
noncentral	specify with a TRUE or FALSE statement whether or not the noncentral approach to sample size planning should be used
standardize	specify with a TRUE or FALSE statement whether or not the regression coefficient will be standardized; default is FALSE
conf_level	desired level of confidence for the computed interval (i.e., 1 - the Type I error rate)
assurance	degree of certainty that the obtained confidence interval will be sufficiently narrow (i.e., the probability that the observed interval will be no larger than desired)

G	the number of generations (i.e., replications) of the simulation within the function
print_iter	specify with a TRUE/FALSE statement if the iteration number should be printed as the simulation within the function runs
filename	Optional path for a comma separated file recording every replication, forwarded to ss_aipe_reg_coef_sensitivity, which does the writing: nothing is written when filename is NULL (the default), and a throwaway run should point it at tempfile(fileext = ".csv").

Details

Direct specification of true_cov_YX and true_cov_XX is necessary, even if one is interested in a single regression coefficient, so that the covariance/correlation structure can be specified when the simulation within the function runs.

Value

A data.frame with columns term and value summarizing the Monte Carlo sensitivity analysis. This function delegates to ss_aipe_reg_coef_sensitivity and inherits its return structure: mean / median / SD summaries of the realized unstandardized regression coefficient, the realized interval widths, and the realized squared multiple correlation coefficient; the proportion of intervals at or below the planning target (pct_ci_less_w); the tail-specific and overall empirical non-coverage rates (pct_ci_miss_low, pct_ci_miss_high, total_type_I_error), all proportions on the 0 to 1 scale; and the input echoes (total_N, p, which_predictor, true_b_j, estimated_b_j, width, conf_level, and, when one was supplied, assurance). See ss_aipe_reg_coef_sensitivity for the full row list.

Note

Note that when the true and estimated covariance structures agree (true_cov_YX equals estimated_cov_YX and true_cov_XX equals estimated_cov_XX), the results are not literally from a sensitivity analysis, rather the function performs a standard simulation study. A simulation study can be helpful in order to determine if the sample size procedure under or overestimates necessary sample size. See ss_aipe_reg_coef_sensitivity in DMAR for more details.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means and Chapter 6 on trend analysis.)

See Also

ss_aipe_reg_coef_sensitivity, ss_aipe_src_sensitivity, ss_aipe_reg_coef, ci_reg_coef

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Sensitivity analysis for an unstandardized regression coefficient
# with two correlated predictors. G is kept small here so the example
# runs quickly; raise G for a stable Monte Carlo summary.
set.seed(113)
Sigma_X <- matrix(c(1, 0.3, 0.3, 1), nrow = 2)
rho_YX  <- c(0.4, 0.3)
cov_YX  <- rho_YX
ss_aipe_rc_sensitivity(
  true_var_Y = 1, true_cov_YX = cov_YX, true_cov_XX = Sigma_X,
  estimated_var_Y = 1, estimated_cov_YX = cov_YX, estimated_cov_XX = Sigma_X,
  which_predictor = 1, w = 0.20, conf_level = 0.95,
  G = 20, print_iter = FALSE
)
```

---

ss_aipe_reg_coef *Sample Size Planning for a Single Regression Coefficient (AIPE)*

Description

Computes the necessary sample size for the confidence interval on a targeted regression coefficient β_j (or its standardized counterpart) in a multiple regression with p predictors to be no wider than a user-specified value. This is the accuracy in parameter estimation (AIPE) framework of Kelley and Maxwell (2003), targeted at a specific coefficient rather than at the omnibus R^2. The `noncentral = TRUE` variant inverts the noncentral t distribution of the standardized b_j under joint multivariate normality of the predictors; the default central-t variant uses a closed form Wald style approximation. Optionally, supplying `assurance` returns the larger N that guarantees the realized width with the specified probability rather than just on average.

Usage

```
ss_aipe_reg_coef(
  rho2_Y_X = NULL,
  rho2_j_X_without_j = NULL,
  p = NULL,
  b_j = NULL,
  width,
  which_width = "Full",
  sigma_Y = 1,
  sigma_X = 1,
```

```
  rho_XX = NULL,
  rho_YX = NULL,
  which_predictor = NULL,
  noncentral = FALSE,
  alpha_lower = NULL,
  alpha_upper = NULL,
  conf_level = 0.95,
  assurance = NULL
)
```

Arguments

rho2_Y_X	Population value of $\rho^2_{Y \cdot X_1,\ldots,X_p}$, the squared multiple correlation of the outcome *Y* with all *p* predictors.
rho2_j_X_without_j	Population value of $\rho^2_{X_j \cdot X_{-j}}$, the squared multiple correlation when the *j*th predictor is regressed on the remaining $p - 1$ predictors. Quantifies the multicollinearity faced by the targeted coefficient.
p	The number of predictor variables.
b_j	The (unstandardized) regression coefficient for the *j*th predictor, the predictor of interest.
width	Desired (full) width of the two-sided confidence interval on β_j.
which_width	Which portion of the confidence interval `width` refers to. Only `"Full"` is currently implemented.
sigma_Y	Population standard deviation of *Y*.
sigma_X	Population standard deviation of the *j*th predictor.
rho_XX	Population correlation matrix for the *p* predictor variables. If supplied with `rho_YX`, `rho2_Y_X` and `rho2_j_X_without_j` are derived from the covariance structure.
rho_YX	Length-*p* vector of population correlations between *Y* and the *p* predictors.
which_predictor	Which of the *p* predictors is the targeted coefficient.
noncentral	If `TRUE`, plans using the exact noncentral *t* sampling distribution of the standardized b_j (Kelley, 2007). If `FALSE` (the default), uses the central *t* approximation. The noncentral path requires `sigma_Y = sigma_X = 1` (*i.e.*, a standardized solution).
alpha_lower	Type I error rate for the lower confidence limit.
alpha_upper	Type I error rate for the upper confidence limit.
conf_level	Confidence level (i.e., $1 - \alpha$, where α is the Type I error rate). Default `0.95`. Mutually exclusive with `alpha_lower` and `alpha_upper`.
assurance	Optional probability with which the realized confidence interval is to be no wider than `width`. When `NULL` (the default), the planning targets the *expected* width.

Details

Calling conventions. The function offers several mutually exclusive ways to supply the population information needed to plan N; the user picks the one most aligned with their available planning values. Specify exactly one of:

- *Covariance structure path.* Supply `rho_XX` and `rho_YX` along with `which_predictor`. The function derives $\rho^2_{Y \cdot X}$ and $\rho^2_{X_j \cdot X_{-j}}$ from the covariance structure and also computes the population b_j for consistency checks.
- *Squared multiple correlations path.* Supply `rho2_Y_X`, `rho2_j_X_without_j`, `p`, and `b_j` directly when the user already has these planning values from prior research and does not need to specify the full covariance structure.
- *Standardized solution.* For the `noncentral = TRUE` path, set `sigma_Y = sigma_X = 1`; the result returns the standardized regression coefficient sample size.

Noncentral vs.\ central planning. The central *t* closed form is fast and adequate at moderate to large *N*; the `noncentral = TRUE` path additionally accounts for the noncentral *t* sampling distribution of the standardized b_j and is preferred when planning at small to moderate *N* or when reporting planning that will be matched against the noncentral CI from `ci_reg_coef`.

Value

A 1-row `data.frame` with columns `term` and `value`. The `term` value is `"necessary_N"` and `value` is the necessary total sample size *N* given the input specifications.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means and Chapter 6 on trend analysis.)

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

See Also

ss_aipe_reg_coef_sensitivity, ci_nc_t

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# 1. Covariance structure path: supply the population correlation
#    matrix and the population YX cross-correlations. Five predictors
#    in an exchangeable structure (all pairwise correlations 0.5,
#    all Y-X correlations 0.3).
rho_YX <- c(.3, .3, .3, .3, .3)
rho_XX <- rbind(c(1, .5, .5, .5, .5), c(.5, 1, .5, .5, .5),
                c(.5, .5, 1, .5, .5), c(.5, .5, .5, 1, .5),
                c(.5, .5, .5, .5, 1))

# Closed-form (central t) planning, targeting the first predictor's
# standardized coefficient.
ss_aipe_reg_coef(width = .10, which_width = "Full",
                 sigma_Y = 1, sigma_X = 1,
                 rho_XX = rho_XX, rho_YX = rho_YX,
                 which_predictor = 1, noncentral = FALSE,
                 conf_level = .95)

# Adding assurance (.85): the realized CI width will be no larger than
# 0.10 in 85 percent of replications. Required N grows accordingly.
ss_aipe_reg_coef(width = .10, which_width = "Full",
                 sigma_Y = 1, sigma_X = 1,
                 rho_XX = rho_XX, rho_YX = rho_YX,
                 which_predictor = 1, noncentral = FALSE,
                 conf_level = .95, assurance = .85)

# Exact noncentral t planning. Required N differs at small to
# moderate samples.
ss_aipe_reg_coef(width = .10, which_width = "Full",
                 sigma_Y = 1, sigma_X = 1,
                 rho_XX = rho_XX, rho_YX = rho_YX,
                 which_predictor = 1, noncentral = TRUE,
                 conf_level = .95)

# 2. Squared multiple correlations path: when the user has planning
#    values for rho^2_Y.X and rho^2_j.X_-j directly (e.g., from a
#    prior power analysis), without specifying the full covariance
#    structure. b_j is required on this path.
ss_aipe_reg_coef(rho2_Y_X = 0.30, rho2_j_X_without_j = 0.20,
                 p = 5, b_j = 0.25,
                 width = .15, which_width = "Full",
                 sigma_Y = 1, sigma_X = 1,
                 noncentral = FALSE, conf_level = .95)
```

ss_aipe_reg_coef_sensitivity *Sensitivity Analysis for Sample Size Planning From the Accuracy in Parameter Estimation Perspective for the (Standardized and Unstandardized) Regression Coefficient*

Description

This function performs a sensitivity analysis when planning sample size from the Accuracy in Parameter Estimation Perspective for the standardized or unstandardized regression coefficient.

Usage

```
ss_aipe_reg_coef_sensitivity(
  true_var_Y = NULL,
  true_cov_YX = NULL,
  true_cov_XX = NULL,
  estimated_var_Y = NULL,
  estimated_cov_YX = NULL,
  estimated_cov_XX = NULL,
  specified_N = NULL,
  which_predictor = 1,
  w = NULL,
  noncentral = FALSE,
  standardize = FALSE,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
  print_iter = TRUE,
  filename = NULL
)
```

Arguments

true_var_Y	Population variance of the dependent variable (Y)
true_cov_YX	Population covariances vector between the p predictor variables and the dependent variable (Y)
true_cov_XX	Population covariance matrix of the p predictor variables
estimated_var_Y	Estimated variance of the dependent variable (Y)
estimated_cov_YX	Estimated covariances vector between the p predictor variables and the dependent variable (Y)
estimated_cov_XX	Estimated Population covariance matrix of the p predictor variables

specified_N	Directly specified sample size (instead of planning one from the estimated covariance structure)
which_predictor	Identifies which of the p predictors is of interest
w	desired Confidence interval width for the regression coefficient of interest
noncentral	Specify with a TRUE or FALSE statement whether or not the noncentral approach to sample size planning should be used
standardize	Specify with a TRUE or FALSE statement whether or not the regression coefficient will be standardized
conf_level	Desired level of confidence for the computed interval (i.e., 1 - the Type I error rate)
assurance	Degree of certainty that the obtained confidence interval will be sufficiently narrow
G	The number of generations (i.e., replications) of the simulation within the function
print_iter	Specify with a TRUE/FALSE statement if the iteration number should be printed as the simulation within the function runs
filename	Optional path of a CSV file to receive the per-replication results (the coefficient estimate, its confidence limits, the observed R^2, the standard error, and the t statistic), appended when the file already exists and created otherwise; the default NULL writes nothing, and a throwaway run that wants the file should point it at tempfile(fileext = ".csv").

Details

Direct specification of true_cov_YX and true_cov_XX is necessary, even if one is interested in a single regression coefficient, so that the covariance/correlation structure can be specified when the simulation within the function runs.

Value

A data.frame with columns term and value summarizing the Monte Carlo sensitivity analysis across G replications. The term entries are: mean_b_j, median_b_j, sd_b_j (summaries of the realized regression-coefficient point estimates); mean_ci_width, median_ci_width, sd_ci_width (summaries of the realized interval widths); pct_ci_less_w (proportion of intervals at or below the planning target w); pct_ci_miss_low and pct_ci_miss_high (tail-specific empirical non-coverage of the population coefficient); total_type_I_error (overall empirical non-coverage, the sum of the two tails); mean_R2, median_R2, sd_R2 (summaries of the realized squared multiple correlation coefficient); and the input echoes total_N (the sample size evaluated), p, which_predictor, true_b_j and estimated_b_j (the population and planning values of the targeted coefficient implied by the supplied covariance structures), width, conf_level, and assurance (present only when an assurance was supplied). The proportion and Type I error rows are proportions on the 0 to 1 scale, not percentages, so total_type_I_error is the sum of pct_ci_miss_low and pct_ci_miss_high.

Note

Note that when the true and estimated covariance structures agree (`true_cov_YX` equals `estimated_cov_YX` and `true_cov_XX` equals `estimated_cov_XX`), the results are not literally from a sensitivity analysis, rather the function performs a standard simulation study. A simulation study can be helpful in order to determine if the sample size procedure under or overestimates necessary sample size.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means and Chapter 6 on trend analysis.)

See Also

`ss_aipe_reg_coef`, `ci_reg_coef`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Sensitivity analysis for an unstandardized regression coefficient
# with two predictors at a modest R squared. The Monte Carlo loop is
# run with a small number of generations (G) here so the example is
# fast; use a larger G (for example G = 1000) in real applications.
set.seed(113)
Sigma_X <- matrix(c(1, 0.3, 0.3, 1), nrow = 2)
cov_YX <- c(0.4, 0.3)
ss_aipe_reg_coef_sensitivity(
  true_var_Y = 1, true_cov_YX = cov_YX, true_cov_XX = Sigma_X,
  estimated_var_Y = 1, estimated_cov_YX = cov_YX, estimated_cov_XX = Sigma_X,
  which_predictor = 1, w = 0.20, conf_level = 0.95,
  G = 100, print_iter = FALSE
)
```

`ss_aipe_reliability` *Sample Size Planning for Accuracy in Parameter Estimation for Reliability Coefficients*

Description

Computes the necessary sample size for the confidence interval on a population reliability coefficient (coefficient alpha or coefficient omega, depending on the assumed measurement model) to have expected width no larger than `width`, or (when `assurance` is supplied) to be no wider than `width` with the specified probability. This is the accuracy in parameter estimation (AIPE) counterpart to power based planning for reliability and is the companion of `reliability`. The closed form planner uses the Terry and Kelley (2012) formulas under the user-selected measurement model and confidence interval type; when `assurance` is supplied, the function follows the closed form with an internal Monte Carlo simulation to find the smallest *N* delivering the requested assurance.

Coefficient alpha (Guttman, 1945; subsequently popularized by Cronbach, 1951) is returned for the parallel and tau equivalent (*i.e.*, the so called True Score) measurement models; coefficient omega (McDonald, 1999) is returned for the congeneric model.

Usage

```
ss_aipe_reliability(
  model = NULL,
  type = NULL,
  width = NULL,
  S = NULL,
  conf_level = 0.95,
  assurance = NULL,
  data = NULL,
  i = NULL,
  cor_est = NULL,
  lambda = NULL,
  psi_square = NULL,
  initial_iter = 500,
  final_iter = 5000,
  start_ss = NULL,
  verbose = FALSE
)
```

Arguments

`model`	The measurement model assumed for the population. Accepts (case-sensitive aliases shown in parentheses): `"Parallel"` (`"parallel"`, `"SB"`, `"Spearman Brown"`, `"Spearman-Brown"`, `"sb"`) for the strictly parallel items model; `"True Score"` (`"True Score Equivalent"`, `"True-Score Equivalent"`, `"Equivalent"`, `"Tau Equivalent"`, `"tau-equivalent"`, `"Tau-Equivalent"`, `"True-Score"`, `"true-score"`, `"true score"`, `"Cronbach"`, `"cronbach"`, `"Chronbach"`, `"alpha"`)

	for the tau equivalent model (in which case the function plans for coefficient alpha); or "Congeneric" ("congeneric", "omega", "Omega") for the congeneric model (in which case the function plans for coefficient omega).
type	The method used to construct the confidence interval on the reliability coefficient: either "Factor Analytic" (McDonald, 1999), available for all three measurement models, or "Normal Theory" (van Zyl, Neudecker, & Nel, 2000), available for the parallel and tau equivalent models only.
width	The desired full width of the two-sided confidence interval.
S	A symmetric population covariance (or correlation) matrix among the items, used to imply the population reliability and its sampling distribution.
conf_level	Confidence level (i.e., $1 - \alpha$, where α is the Type I error rate). Default 0.95.
assurance	Optional probability with which the realized interval is to be no wider than width. When NULL (the default), the planner targets the *expected* width; when supplied (e.g., 0.80, 0.85, 0.95), the function follows the closed form with a Monte Carlo search.
data	A data set from which the population covariance matrix should be inferred.
i	Number of items.
cor_est	The presumed inter-item correlation. One value for the parallel and tau equivalent models.
lambda	Vector of population factor loadings.
psi_square	Vector of population unique (error) variances.
initial_iter	Number of Monte Carlo iterations used in the initial assurance search.
final_iter	Number of Monte Carlo iterations used in the final assurance verification.
start_ss	Optional starting sample size for the iterative assurance search.
verbose	If TRUE, prints the current sample size and empirical assurance at each step of the Monte Carlo search.

Details

The Monte Carlo assurance search simulates covariance matrices from the population implied by the inputs and, at each candidate sample size, computes the realized confidence interval width with the same machinery the estimation side of the package uses. For type = "Factor Analytic" the single-factor model is fit by maximum likelihood (equal loadings for the parallel and tau equivalent models, free loadings for the congeneric model) and the interval is the delta method Wald interval on the model implied reliability, the interval of reliability_omega(denominator = "model_implied",ci_method = "ml") and of reliability_alpha(estimator = "model_implied",ci_method = "ml"). For type = "Normal Theory" the interval uses the van Zyl, Neudecker, and Nel (2000) closed form standard error for the tau equivalent model and its compound symmetry simplification for the parallel model, the same closed form behind reliability_alpha(ci_method = "ml"). The congeneric model has no normal theory form, so type = "Normal Theory" is an error there.

The cost of the assurance search depends on the interval. The normal theory intervals are closed forms, so a search at the default initial_iter = 500 and final_iter = 5000 finishes in seconds. With the factor analytic interval a one factor model is fit at every Monte Carlo iteration and at every candidate sample size the search visits, so a congeneric plan with an assurance runs for tens of

seconds even at `initial_iter = 50` and `final_iter = 200` and for many minutes at the defaults. The examples on this page therefore stop at the closed form for the congeneric plan; the assurance version is the same call with `assurance` supplied (for example, `assurance = .80`), and `set.seed()` before the call makes the search reproducible.

Value

A `data.frame` with columns `term` and `value`. Without `assurance` the data frame has a single row, `"necessary_N"`, giving the necessary *N*. With `assurance` supplied, the data frame has five rows: `"necessary_N"` (necessary *N*), `"width"` (echo of the target width), `"specified_assurance"` (echo of the requested probability), `"empirical_assurance"` (the assurance achieved at the returned *N* in the Monte Carlo verification), and `"final_iter"` (number of Monte Carlo iterations used).

Warning

In some conditions the factor analytic model fit by `cfa_1` (via **lavaan**) may fail to converge, and you may see a non-convergence message from **lavaan**. The Monte Carlo assurance search treats a non-converged iteration as missing and continues, so a few such messages do not invalidate the result. Frequent non-convergence usually means the model is poorly determined by the data, for example because of a small sample size, a low number of iterations, or a poorly behaved covariance matrix.

Note

Not all of the items can be entered into the function to represent the population values. For example, either 'data' can be used, or `S`, or `i`, `cor_est`, and `psi_square`, or `i`, `lambda`, and `psi_square`. With a large number of iterations (`final_iter`) this function may take considerable time.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Cheng, Y. (2012). Estimation of and confidence interval formation for reliability coefficients of homogeneous measurement instruments. *Methodology, 8*, 39–50. doi:10.1027/1614-2241/a000036

Kelley, K., & Pornprasertmanit, S. (2016). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for composite measures. *Psychological Methods, 21*, 69–92. doi:10.1037/a0040086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

McDonald, R. P. (1999). *Test theory: A unified treatment*. Mahwah, NJ: Lawrence Erlbaum Associates.

Terry, L. J., & Kelley, K. (2012). Sample size planning for composite reliability coefficients: Accuracy in parameter estimation via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 371–401. doi:10.1111/j.20448317.2011.02030.x

van Zyl, J. M., Neudecker, H., & Nel, D. G. (2000). On the distribution of the maximum likelihood estimator of Cronbach's alpha. *Psychometrika, 65*(3), 271–280. doi:10.1007/BF02296146

See Also

reliability, cfa_1

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Expected confidence interval width (closed form, no Monte Carlo search).
ss_aipe_reliability(model = "Parallel", type = "Normal Theory", width = .1,
  i = 6, cor_est = .3, psi_square = .2, conf_level = .95, assurance = NULL)

# The assurance cases run a Monte Carlo search. initial_iter = 50 and
# final_iter = 200 keep the examples quick; a reported plan deserves the
# defaults of 500 and 5000, and set.seed() makes the search reproducible.
set.seed(113)

# Same population, now targeting an assurance.
ss_aipe_reliability(model = "Parallel", type = "Normal Theory", width = .1,
  i = 6, cor_est = .3, psi_square = .2, conf_level = .95, assurance = .85,
  initial_iter = 50, final_iter = 200)

# The true score (tau equivalent) model takes psi_square as a vector of
# length i (number of items) while cor_est stays a single value.
ss_aipe_reliability(model = "True Score", type = "Normal Theory",
  width = .1, i = 5, cor_est = .3, psi_square = c(.2, .3, .3, .2, .3),
  conf_level = .95, assurance = .85, initial_iter = 50, final_iter = 200)

# Congeneric model, planned from the item loadings and error variances rather
# than from a single correlation. With assurance = NULL the necessary N comes
# from the closed form expected width evaluated at the implied population
# correlation matrix, so type does not enter the answer; type selects the
# interval that the Monte Carlo assurance search evaluates. Adding an
# assurance to this plan fits a one factor model at every Monte Carlo
# iteration and is the slow case, so the page stops at the closed form
# (see Details).
ss_aipe_reliability(model = "Congeneric", type = "Factor Analytic", width = .15,
  i = 4, lambda = c(.8, .7, .7, .8), psi_square = c(.4, .5, .5, .4),
  conf_level = .95, assurance = NULL)

# Planning from a presumed population correlation matrix among the items.
pop_mat <- rbind(
  c(1.0000000, 0.3813850, 0.4216370, 0.3651484, 0.4472136),
  c(0.3813850, 1.0000000, 0.4020151, 0.3481553, 0.4264014),
  c(0.4216370, 0.4020151, 1.0000000, 0.3849002, 0.4714045),
  c(0.3651484, 0.3481553, 0.3849002, 1.0000000, 0.4082483),
  c(0.4472136, 0.4264014, 0.4714045, 0.4082483, 1.0000000))
ss_aipe_reliability(model = "True Score", type = "Normal Theory", width = .15,
  S = pop_mat, conf_level = .95, assurance = .85, initial_iter = 50,
  final_iter = 200)
```

`ss_aipe_reliability_sensitivity`
Sensitivity Analysis for Sample Size Planning From the AIPE Perspective for a Reliability Coefficient

Description

Quantifies how much misspecification of the population reliability coefficient distorts an AIPE-based sample size plan for the composite-score reliability. On each replication the function simulates an $n \times i$ item-by-subject data matrix from a single-factor parallel-tests model whose population reliability of the sum score equals `true_reliability`, fits the requested estimator (alpha or omega) via the corresponding `reliability_*` function with the supplied `ci_method`, and records the realized reliability estimate and its confidence interval.

Population model. Each item has a single common-factor loading and uncorrelated unique error. With per-item variance normalized to 1, the loading and unique variance are chosen so the Cronbach-style sum-score reliability equals `true_reliability`:

$$\lambda^2 \;=\; \frac{\rho}{i(1-\rho)+\rho}, \qquad \psi^2 \;=\; 1-\lambda^2,$$

where ρ =`true_reliability` and i is the item count. Item scores are $y_{ij} = \lambda T_i + e_{ij}$, with $T_i \sim N(0,1)$ and $e_{ij} \sim N(0,\psi^2)$.

Usage

```
ss_aipe_reliability_sensitivity(
  true_reliability = NULL,
  estimated_reliability = NULL,
  i,
  width,
  specified_N = NULL,
  estimator = c("alpha", "omega"),
  ci_method = NULL,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

`true_reliability`
: Population reliability coefficient (in $[0, 1)$).

`estimated_reliability`
: Reliability used to plan the study; the function passes the implied lambda / psi^2 to `ss_aipe_reliability`.

i	Number of items in the composite.
width	Desired full width of the CI on reliability.
specified_N	Sample size to evaluate (incompatible with estimated_reliability).
estimator	One of "alpha" (default; coefficient alpha via reliability_alpha) or "omega" (composite reliability via reliability_omega). For a parallel-tests population the two coincide; differences in sample estimates reflect estimator-specific finite-sample bias and CI behavior.
ci_method	CI method passed to the estimator. Default "bonett" for alpha and "mlr" for omega.
conf_level	Confidence level (default 0.95).
assurance	Optional assurance probability passed to the planner.
G	Number of Monte Carlo replications.
print_iter	Logical.
filename	Optional path for a comma separated file recording every replication (the sample reliability estimate, the two confidence limits, the interval width, and two indicators of whether the interval missed true_reliability below or above): nothing is written when filename is NULL (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at tempfile(fileext = ".csv").

Value

A data.frame with rows for mean / median / SD of the realized reliability and CI width, the proportion of intervals at or below width, tail-specific and overall non-coverage of true_reliability, and the input echoes, including assurance (present only when an assurance was supplied).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Pornprasertmanit, S. (2016). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for composite measures. *Psychological Methods, 21*, 69–92. doi:10.1037/a0040086

Terry, L. J., & Kelley, K. (2012). Sample size planning for composite reliability coefficients: Accuracy in parameter estimation via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 371–401. doi:10.1111/j.20448317.2011.02030.x

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

ss_aipe_reliability, reliability_alpha, reliability_omega

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(),
ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_icc_sensitivit
ss_aipe_indirect_effect(), ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(),
ss_aipe_omega_squared(), ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(),
ss_aipe_partial_r_sensitivity(), ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(),
ss_aipe_semipartial_r(), ss_aipe_semipartial_r_sensitivity()

Examples

```
# Reduced Monte Carlo sweep (small G) so the example runs quickly;
# raise G for a production sensitivity analysis.
set.seed(113)
ss_aipe_reliability_sensitivity(
  true_reliability      = 0.80,
  estimated_reliability = 0.80,
  i = 4, width = 0.15,
  estimator = "alpha",
  G = 20, print_iter = FALSE
)
```

---

ss_aipe_rmsea *Sample Size Planning for RMSEA in SEM*

Description

Sample size planning for the population root mean square error of approximation (RMSEA) from the accuracy in parameter estimation (AIPE) perspective. The sample size is planned so that the expected width of a confidence interval for the population RMSEA is no larger than desired.

Usage

```
ss_aipe_rmsea(RMSEA, df, width, conf_level = 0.95)
```

Arguments

RMSEA	The input RMSEA value
df	Degrees of freedom of the model
width	Desired confidence interval width
conf_level	Desired confidence level (e.g., .90, .95, .99, etc.)

Value

Returns the necessary total sample size in order to achieve the desired degree of accuracy (i.e., the sufficiently narrow confidence interval).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Lai, K. (2011). Accuracy in parameter estimation for the root mean square error of approximation: Sample size planning for narrow confidence intervals. *Multivariate Behavioral Research, 46*, 1–32. doi:10.1080/00273171.2011.543027

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

ci_rmsea

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
ss_aipe_rmsea(RMSEA = .035, df = 50, width = .05, conf_level = .95)
```

---

ss_aipe_rmsea_sensitivity

A Priori Monte Carlo Simulation for Sample Size Planning for RMSEA in SEM

Description

Conduct a priori Monte Carlo simulation to empirically study the effects of (mis)specifications of input information on the calculated sample size. The sample size is planned so that the expected width of a confidence interval for the population RMSEA is no larger than desired. Random data are generated from the true covariance matrix but fit to the proposed model, whereas the sample size is calculated based on the input covariance matrix and proposed model.

Usage

```
ss_aipe_rmsea_sensitivity(
  width,
  model,
  Sigma,
  N = NULL,
  conf_level = 0.95,
  G = 200,
  filename = NULL,
  ...
)
```

Arguments

width	desired confidence interval width for the population RMSEA.
model	the model the researcher proposes, which may or may not be the true model, written in **lavaan** model syntax (see model.syntax). The observed variable names in the model must match the row and column names of Sigma.
Sigma	the true population covariance matrix, which is used to generate random data for the simulation study. The row and column names of Sigma must match the observed variables in model.
N	if N is specified, random samples of the specified size are generated. Otherwise the sample size is calculated with the sample size planning method so that the expected width of a confidence interval for the population RMSEA is no larger than width.
conf_level	confidence level (i.e., 1 - the Type I error rate).
G	number of replications in the Monte Carlo simulation.
filename	an optional path for a comma separated file recording every converged replication (its index, the RMSEA estimate, the two confidence limits, and the interval width): nothing is written when filename is NULL (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at tempfile(fileext = ".csv").
...	additional arguments passed to sem when fitting the model (for example estimator or missing).

Details

This function implements the sample size planning method proposed in Kelley and Lai (2011). It uses sem to fit the proposed model to the population covariance matrix, which recovers the population RMSEA (the model misspecification) and the model degrees of freedom, and to fit the model to each simulated sample, and it uses ci_rmsea to construct the confidence interval for the population RMSEA in each replication. The model is specified in **lavaan** syntax, so **lavaan** must be installed.

Earlier versions of this function used the **sem** package to fit the model. The fit is now carried out with **lavaan**, the structural equation modeling backend used throughout DMAR. The population RMSEA is read from lavaan::fitMeasures(), which is computed reliably for the large-sample population fit.

Value

A data.frame with columns term and value summarizing the a priori Monte Carlo study. The term entries are: "mean_rmsea", "median_rmsea", "sd_rmsea" (summaries of the realized RMSEA estimates across the converged replications); "mean_ci_width", "median_ci_width", "sd_ci_width" (summaries of the realized interval widths); "pct_ci_less_w" (proportion of intervals narrower than the target width); "pct_ci_miss_low" and "pct_ci_miss_high" (tail-specific empirical non-coverage of the population RMSEA); "total_type_I_error" (overall empirical non-coverage, the sum of the two tails); and the echoes "suc_rep" (number of converged replications), "total_N" (the N evaluated), "df" (model degrees of freedom), "true_rmsea" (the population RMSEA recovered from fitting model to Sigma), "width", and "conf_level". The proportion rows are on the 0 to 1 scale, not percentages.

Note

Replications in which **lavaan** fails to converge, or for which the RMSEA is undefined, are skipped; the number of converged replications is reported as `suc_rep`. Increase `G` if many replications fail to converge.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cudeck, R., & Browne, M. W. (1992). Constructing a covariance matrix that yields a specified minimizer and a specified minimum discrepancy function value. *Psychometrika, 57*, 357–369. doi:10.1007/BF02295424

Kelley, K., & Lai, K. (2011). Accuracy in parameter estimation for the root mean square error of approximation: Sample size planning for narrow confidence intervals. *Multivariate Behavioral Research, 46*, 1–32. doi:10.1080/00273171.2011.543027

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48*(2), 1–36. doi:10.18637/jss.v048.i02

See Also

`sem`, `ss_aipe_rmsea`, `ci_rmsea`

Examples

```
# True data generating model: two correlated factors, each measured by
# three standardized indicators. The factor correlation is 0.5 and every
# loading is 0.7. The implied population covariance matrix is assembled
# from the loading matrix, the factor correlation matrix, and the
# residual variances.
Lambda <- matrix(0, 6, 2)
Lambda[1:3, 1] <- 0.7
Lambda[4:6, 2] <- 0.7
Phi    <- matrix(c(1, 0.5, 0.5, 1), 2, 2)
Sigma <- Lambda %*% Phi %*% t(Lambda) + diag(1 - 0.7^2, 6)
dimnames(Sigma) <- list(paste0("x", 1:6), paste0("x", 1:6))

# Proposed (misspecified) model: a single common factor.
proposed <- "g =~ x1 + x2 + x3 + x4 + x5 + x6"

# The proposed model is fit once at a very large N to recover the
# population RMSEA, the sample size is planned so that the expected width
# of the 95 percent interval is 0.05, and a fresh sample of that size is
# drawn and fit on every replication. Notice that true_rmsea is about
# 0.20, since a single factor is a poor description of two-factor data,
# that the realized widths sit close to the target, and that
# pct_ci_less_w is near one half, which is what planning for the expected
```

```
# width delivers. G = 20 keeps the example quick; a reported sensitivity
# study deserves the default G = 200 or more.
set.seed(113)
ss_aipe_rmsea_sensitivity(width = 0.05, model = proposed, Sigma = Sigma,
                          G = 20)
```

`ss_aipe_sc` *Sample Size Planning for Accuracy in Parameter Estimation (AIPE) of the Standardized Contrast in ANOVA*

Description

Plans the sample size per group so that the confidence interval for a standardized contrast of means in a fixed effects analysis of variance, the interval computed by `ci_sc`, is sufficiently narrow, an application of the accuracy in parameter estimation (AIPE) approach to the standardized contrast.

Usage

```
ss_aipe_sc(
  psi_standardized,
  c_weights,
  width,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  assurance = NULL,
  ...
)
```

Arguments

`psi_standardized`	Population standardized contrast
`c_weights`	The contrast weights
`width`	The desired full width of the obtained confidence interval
`conf_level`	The desired confidence interval coverage (i.e., 1 - Type I error rate). Default is `.95`, which gives a symmetric two-sided interval. Specify either `conf_level` or both of `alpha_lower` and `alpha_upper`, not both.
`alpha_lower`	Lower-tail Type I error rate, used to plan an asymmetric confidence interval. When supplied together with `alpha_upper`, the planned interval has lower-tail probability `alpha_lower` and upper-tail probability `alpha_upper`. Set `conf_level = NULL` when supplying these.
`alpha_upper`	Upper-tail Type I error rate, used together with `alpha_lower` to plan an asymmetric confidence interval.

assurance — Parameter to ensure that the obtained confidence interval width is narrower than the desired width with a specified degree of certainty (must be NULL or between zero and unity)

... — Allows one to potentially include parameter values for inner functions

Value

necessary_n_per_group — Necessary sample size *per group*

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*(4), 532–574. doi:10.1177/0013164401614002

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Kelley, K. (2005). The effects of nonnormal distributions on confidence intervals around the standardized mean difference: Bootstrap and parametric confidence intervals, *Educational and Psychological Measurement, 65*, 51–69. doi:10.1177/0013164404264850

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

ci_sc, ci_nc_t, ss_aipe_c

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Suppose the population standardized contrast is believed to be .6
# in some 5-group ANOVA model. The researcher is interested in comparing
# the average of means of group 1 and 2 with the average of group 3 and 4.
```

```
# To calculate the necessary sample size per group such that the width
# of 95 percent confidence interval of the standardized
# contrast is, with 90 percent assurance, no wider than .4:

ss_aipe_sc(psi_standardized=.6, c_weights=c(.5, .5, -.5, -.5, 0), width=.4, assurance=.90)

# Asymmetric confidence interval: most of the alpha goes in the upper tail
# (e.g., when a one-sided concern dominates). Pass alpha_lower and
# alpha_upper instead of conf_level.
ss_aipe_sc(psi_standardized = .6, c_weights = c(.5, .5, -.5, -.5, 0), width = .4,
           conf_level = NULL, alpha_lower = .01, alpha_upper = .04)
```

`ss_aipe_sc_ancova` *Sample Size Planning From the AIPE Perspective for Standardized ANCOVA Contrasts*

Description

Sample size planning from the accuracy in parameter estimation (AIPE) perspective for standardized ANCOVA contrasts.

Usage

```
ss_aipe_sc_ancova(
  psi = NULL,
  sigma_anova = NULL,
  sigma_ancova = NULL,
  psi_standardized = NULL,
  ratio = NULL,
  rho = NULL,
  divisor = "s_ancova",
  c_weights,
  width,
  conf_level = 0.95,
  alpha_lower = NULL,
  alpha_upper = NULL,
  assurance = NULL,
  ...
)
```

Arguments

`psi`	The population unstandardized ANCOVA (adjusted) contrast
`sigma_anova`	The population error standard deviation of the ANOVA model
`sigma_ancova`	The population error standard deviation of the ANCOVA model
`psi_standardized`	The population standardized ANCOVA (adjusted) contrast

ratio	The ratio of sigma_ancova over sigma_anova
rho	The population correlation coefficient between the response and the covariate
divisor	Which error standard deviation to be used in standardizing the contrast; the value can be either "s_ancova" or "s_anova"
c_weights	Contrast weights
width	The desired full width of the obtained confidence interval
conf_level	The desired confidence interval coverage (i.e., 1 - Type I error rate). Default is .95, which gives a symmetric two-sided interval. Specify either conf_level or both of alpha_lower and alpha_upper, not both.
alpha_lower	Lower-tail Type I error rate, used to plan an asymmetric confidence interval. When supplied together with alpha_upper, the planned interval has lower-tail probability alpha_lower and upper-tail probability alpha_upper. Set conf_level = NULL when supplying these.
alpha_upper	Upper-tail Type I error rate, used together with alpha_lower to plan an asymmetric confidence interval.
assurance	Parameter to ensure that the obtained confidence interval width is narrower than the desired width with a specified degree of certainty (must be NULL or between zero and unity)
...	Allows one to potentially include parameter values for inner functions

Details

The sample size planning method this function is based on is developed in the context of simple (i.e., one-response-one-covariate) ANCOVA model and randomized design (i.e., same population covariate mean across groups).

An ANCOVA contrast can be standardized in at least two ways: (a) divided by the error standard deviation of the ANOVA model, (b) divided by the error standard deviation of the ANCOVA model. This function can be used to analyze both types of standardized ANCOVA contrasts.

Not all of the effect size arguments need to be specified. When divisor="s_ancova" the input is either (a) psi_standardized, or (b) psi (the unstandardized ANCOVA contrast) and sigma_ancova. When divisor="s_anova", the valid input combinations are (a) psi_standardized and ratio; (b) psi_standardized and rho; or (c) psi, sigma_anova, and sigma_ancova.

Value

A 1-row data.frame with columns term and value. The term is "necessary_n_per_group" and value is the per-group sample size needed for the planned ANCOVA contrast.

Note

When divisor="s_anova" and the argument assurance is specified, the necessary sample size *per group* returned by the function with assurance specified is slightly underestimated. The method to obtain exact sample size in the above situation has not been developed yet. A practical solution is to use the sample size returned as the starting value to conduct a priori Monte Carlo simulations with function ss_aipe_sc_ancova_sensitivity, as discussed in Lai & Kelley (2012).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9.)

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

ss_aipe_sc, ss_aipe_sc_ancova_sensitivity

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
ss_aipe_sc_ancova(psi_standardized = .8, width = .5, c_weights = c(.5, .5, 0, -1))

ss_aipe_sc_ancova(psi_standardized = .8, ratio = .6, width = .5,
                  c_weights = c(.5, .5, 0, -1), divisor = "s_anova")

ss_aipe_sc_ancova(psi_standardized = .5, rho = .4, width = .3,
                c_weights = c(.5, .5, 0, -1), divisor = "s_anova")
```

---

ss_aipe_sc_ancova_sensitivity

Sensitivity Analysis for the Sample Size Planning Method for Standardized ANCOVA Contrast

---

Description

Sensitivity analysis for the sample size planning method with the goal to obtain sufficiently narrow confidence intervals for standardized ANCOVA complex contrasts.

Usage

```
ss_aipe_sc_ancova_sensitivity(
  true_psi = NULL,
  estimated_psi = NULL,
  c_weights,
  desired_width = NULL,
  n_per_group = NULL,
  mu_x = 0,
  sigma_x = 1,
  rho,
  divisor = "s_ancova",
  assurance = NULL,
  conf_level = 0.95,
  G = 10000,
  print_iter = TRUE,
  filename = NULL,
  ...
)
```

Arguments

true_psi	the population standardized ANCOVA contrast
estimated_psi	the estimated standardized ANCOVA contrast
c_weights	the contrast weights
desired_width	the desired full width of the obtained confidence interval
n_per_group	selected sample size to use in order to determine distributional properties of a given value of sample size
mu_x	the population mean for the covariate
sigma_x	the population standard deviation of the covariate
rho	the population correlation coefficient between the response and the covariate
divisor	which error standard deviation to be used in standardizing the contrast; the value can be either "s_ancova" or "s_anova"
assurance	parameter to ensure that the obtained confidence interval width is narrower than the desired width with a specified degree of certainty (must be NULL or between zero and unity)
conf_level	the desired confidence interval coverage, (i.e., 1 - Type I error rate)
G	number of generations (i.e., replications) of the simulation
print_iter	to print the current value of the iterations
filename	Optional path of a CSV file to receive the per-replication results (the observed standardized contrast, the full and one-sided interval widths, the tail and overall misses, and the confidence limits), appended when the file already exists and created otherwise; the default NULL writes nothing, and a throwaway run that wants the file should point it at tempfile(fileext = ".csv").
...	allows one to potentially include parameter values for inner functions

Details

The sample size planning method this function is based on is developed in the context of simple (i.e., one-response-one-covariate) ANCOVA model and randomized design (i.e., same population covariate mean across groups).

An ANCOVA contrast can be standardized in at least two ways: (a) divided by the error standard deviation of the ANOVA model, (b) divided by the error standard deviation of the ANCOVA model. This function can be used to analyze both types of standardized ANCOVA contrasts.

The population mean and standard deviation of the covariate does not affect the sample size planning procedure; they can be specified as any values that are considered as reasonable by the user.

Value

A `data.frame` with columns `term` and `value` summarizing the Monte Carlo sensitivity analysis across `G` replications. The `term` entries are: `mean_psi`, `median_psi`, `sd_psi` (summaries of the realized standardized ANCOVA contrast); `mean_ci_width`, `median_ci_width`, `sd_ci_width` (summaries of the full interval widths); `mean_ci_width_lower` and `mean_ci_width_upper` (mean one-sided widths, measured from the observed contrast to each limit); `pct_ci_less_w` (proportion of intervals at or below the target width); `pct_ci_miss_low` and `pct_ci_miss_high` (tail-specific empirical non-coverage of `true_psi`); `total_type_I_error` (overall empirical non-coverage, the sum of the two tails); and the input echoes `n_per_group`, `total_N`, `true_psi`, `estimated_psi` (NA when `n_per_group` was supplied instead), `rho`, `width`, `conf_level`, and `assurance` (present only when an assurance was supplied). The proportion and Type I error rows are proportions on the 0 to 1 scale, not percentages.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9.)

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

ss_aipe_sc_ancova, ss_aipe_sc_sensitivity

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Sensitivity analysis for a standardized ANCOVA contrast across
# three groups, contrast (-1, 0, 1), a covariate-outcome correlation
# of 0.4, and a planning target width of 0.5. G = 50 keeps the
# example quick; a reported sensitivity analysis deserves the default of
# G = 10000 replications.
set.seed(113)
ss_aipe_sc_ancova_sensitivity(
  true_psi = 0.5, estimated_psi = 0.5,
  c_weights = c(-1, 0, 1),
  desired_width = 0.5, rho = 0.4,
  conf_level = 0.95, G = 50, print_iter = FALSE
)
```

---

ss_aipe_sc_sensitivity

Sensitivity Analysis for Sample Size Planning for the Standardized ANOVA Contrast From the Accuracy in Parameter Estimation (AIPE) Perspective

---

Description

Performs a sensitivity analysis when planning sample size from the Accuracy in Parameter Estimation (AIPE) Perspective for the standardized ANOVA contrast.

Usage

```
ss_aipe_sc_sensitivity(
  true_psi = NULL,
  estimated_psi = NULL,
  c_weights,
  desired_width = NULL,
  n_per_group = NULL,
  assurance = NULL,
  conf_level = 0.95,
  G = 10000,
  print_iter = TRUE,
  filename = NULL
)
```

Arguments

true_psi	population standardized contrast
estimated_psi	estimated standardized contrast
c_weights	the contrast weights
desired_width	the desired full width of the obtained confidence interval
n_per_group	selected sample size to use in order to determine distributional properties of at a given value of sample size
assurance	parameter to ensure that the obtained confidence interval width is narrower than the desired width with a specified degree of certainty (must be NULL or between zero and unity)
conf_level	the desired confidence interval coverage, (i.e., 1 - Type I error rate)
G	number of generations (i.e., replications) of the simulation
print_iter	to print the current value of the iterations
filename	an optional path for a comma separated file recording every replication (the realized standardized contrast, the full and the two one-sided interval widths, the three non-coverage indicators, and the two confidence limits): nothing is written when filename is NULL (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at tempfile(fileext = ".csv")

Value

A data.frame with columns term and value summarizing the Monte Carlo sensitivity analysis across the G replications. The term entries are: mean_psi, median_psi, sd_psi (summaries of the realized standardized contrast); mean_ci_width, median_ci_width, sd_ci_width (summaries of the full interval widths); mean_ci_width_lower and mean_ci_width_upper (mean one-sided widths, measured from the observed contrast to each limit); pct_ci_less_w (proportion of intervals at or below the target width); pct_ci_miss_low and pct_ci_miss_high (tail-specific empirical non-coverage of true_psi); total_type_I_error (overall empirical non-coverage, the sum of the two tails); and the input echoes n_per_group, total_N, true_psi, estimated_psi (NA when n_per_group was supplied instead), width, conf_level, and assurance (present only when an assurance was supplied). The proportion and Type I error rows are proportions on the 0 to 1 scale, not percentages.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*(4), 532–574. doi:10.1177/0013164401614002

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in Parameter Estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

`ss_aipe_sc`, `ss_aipe_c`, `ci_nc_t`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Sensitivity analysis for a standardized three-group ANOVA contrast
# (-1, 0, 1) at psi = 0.5 and target full width 0.40. G is kept small
# here so the example runs quickly; raise it for a stable sweep.
set.seed(113)
ss_aipe_sc_sensitivity(
  true_psi = 0.5, estimated_psi = 0.5,
  c_weights = c(-1, 0, 1),
  desired_width = 0.40,
  conf_level = 0.95, G = 50, print_iter = FALSE
)
```

`ss_aipe_sem_path` *Sample Size Planning for SEM Targeted Effects*

Description

Plan sample size for structural equation models so that the confidence interval for the targeted model parameter is sufficiently narrow

Usage

```
ss_aipe_sem_path(
  model,
  Sigma,
  desired_width,
  which_path,
```

```
  conf_level = 0.95,
  assurance = NULL,
  detail = FALSE,
  internal = FALSE,
  ...
)
```

Arguments

model	A single character string giving the free analysis model in lavaan model syntax (see model.syntax). The target path must carry a parameter label so it can be referred to by name, for example "f2 ~ b*f1" labels the structural path b. This is the model that would be fit to the data; its parameters are free, not fixed to population values
Sigma	Estimated population covariance matrix of the observed variables, with row and column names matching the observed variables in model. It is typically obtained from a fully fixed population model via cov_sem
desired_width	Desired confidence interval width for the model parameter of interest
which_path	The parameter label of the targeted path, given as a character string, for example "b" for the path labeled f2 ~ b*f1 in model
conf_level	Confidence level (i.e., 1 - Type I error rate)
assurance	The assurance that the confidence interval obtained in a particular study will be no wider than desired (must be NULL or a value between 0.50 and 1)
detail	if TRUE, additionally print the model parameter names and the observed variable names (the returned table is unchanged)
internal	option to output a list for internal use (for ss_aipe_sem_path_sensitivity)
...	Allows one to potentially pass additional arguments to sem

Details

This function implements the sample size planning methods proposed in Lai and Kelley (2011). It requires **lavaan** to be installed and uses sem to obtain the expected information, that is the asymptotic covariance matrix of the parameter estimates, by fitting the free analysis model to the population covariance matrix Sigma at a very large sample size. The analysis model is written in lavaan model syntax with the targeted path given a parameter label; see model.syntax for the syntax and sem for the fitting machinery. The population covariance matrix Sigma is most naturally produced by cov_sem from a fully fixed population model.

When assurance is supplied, the assurance adjustment is based on a chi square approximation to the sampling variability of the confidence interval width and can undershoot the nominal assurance in finite samples; use ss_aipe_sem_path_sensitivity to check the realized width and coverage at the planned sample size.

Value

A data.frame (a dmar_tbl) with term and value columns whose rows are necessary_N (the planned sample size), path_index (the position of the target path among the model parameters),

and var_theta_j (the population sampling variance of the target path at the planned sample size). The returned table is the same whether or not detail = TRUE. When internal = TRUE a list is returned for use by ss_aipe_sem_path_sensitivity.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Lai, K., & Kelley, K. (2011). Accuracy in parameter estimation for targeted effects in structural equation modeling: Sample size planning for narrow confidence intervals. *Psychological Methods, 16*(2), 127–148. doi:10.1037/a0021764

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48*(2), 1–36. doi:10.18637/jss.v048.i02

See Also

sem, model.syntax, cov_sem, ss_aipe_sem_path_sensitivity

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Population covariance from a fully fixed model (see cov_sem()).
pop_model <- "
  f1 =~ 1*y1 + 0.8*y2 + 0.8*y3
  f2 =~ 1*y4 + 0.8*y5 + 0.8*y6
  f2 ~ 0.5*f1
  f1 ~~ 1*f1
  f2 ~~ 0.75*f2
  y1 ~~ 0.5*y1; y2 ~~ 0.5*y2; y3 ~~ 0.5*y3
  y4 ~~ 0.5*y4; y5 ~~ 0.5*y5; y6 ~~ 0.5*y6
"
Sigma <- cov_sem(pop_model)$sigma_theta

# Free analysis model with the target structural path labeled "b".
analysis_model <- "
  f1 =~ y1 + y2 + y3
  f2 =~ y4 + y5 + y6
  f2 ~ b*f1
"
ss_aipe_sem_path(model = analysis_model, Sigma = Sigma,
                 desired_width = 0.30, which_path = "b")

# That sample size holds the interval to the desired width on average, so
# about half of the studies it plans return a wider one. Adding assurance
# plans for the width to be met in 90 percent of studies instead, at the
```

```
# cost of a larger sample size.
ss_aipe_sem_path(model = analysis_model, Sigma = Sigma,
                 desired_width = 0.30, which_path = "b",
                 assurance = 0.90)
```

`ss_aipe_sem_path_sensitivity` *A Priori Monte Carlo Simulation for Sample Size Planning for SEM Targeted Effects*

Description

Conduct a priori Monte Carlo simulation to empirically study the effects of (mis)specifications of input information on the calculated sample size. Random data are generated from the true covariance matrix but fit to the proposed model, whereas sample size is calculated based on the input covariance matrix and proposed model.

Usage

```
ss_aipe_sem_path_sensitivity(
  model,
  est_Sigma,
  true_Sigma = est_Sigma,
  which_path,
  desired_width,
  N = NULL,
  conf_level = 0.95,
  assurance = NULL,
  G = 100,
  filename = NULL,
  ...
)
```

Arguments

`model`	A single character string giving the free analysis model in lavaan model syntax (see `model.syntax`), the model that would be fit to the data. The target path must carry a parameter label so it can be referred to by name, for example `"f2 ~ b*f1"` labels the structural path b. The model may or may not be the true data generating model
`est_Sigma`	the covariance matrix used to calculate sample size, may or may not be the true covariance matrix. The row names and column names of `est_Sigma` should be the same as the observed variables in `model`
`true_Sigma`	the true population covariance matrix, which will be used to generate random data for the simulation study. The row names and column names of `true_Sigma` should be the same as the observed variables in `model`

which_path	the parameter label of the targeted path, given as a character string, for example "b" for the path labeled f2 ~ b*f1 in model
desired_width	desired confidence interval width for the model parameter of interest
N	the sample size of random data. If it is NULL, it will be determined by the sample size planning method
conf_level	confidence level (i.e., 1- Type I error rate)
assurance	the assurance that the confidence interval obtained in a particular study will be no wider than desired (must be NULL or a value between 0.50 and 1)
G	number of replications in the Monte Carlo simulation
filename	an optional path for a comma separated file recording every replication (the estimate of the targeted path, its standard error, the two confidence limits, and the interval width): nothing is written when filename is NULL (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at tempfile(fileext = ".csv")
...	allows one to potentially include parameter values for inner functions

Details

This function implements the sample size planning methods proposed in Lai and Kelley (2011). It calls ss_aipe_sem_path to plan the sample size and to identify the targeted path, then fits the analysis model to data simulated from true_Sigma with sem. The analysis model is written in lavaan model syntax with the targeted path given a parameter label; see model.syntax for the syntax and sem for the fitting machinery. The population covariance matrices are most naturally produced by cov_sem from a fully fixed population model. This function requires **lavaan** and **MASS** to be installed.

Value

A data.frame with columns term and value summarizing the a priori Monte Carlo study. The term entries are: "mean_path", "median_path", "sd_path" (summaries of the realized estimates of the targeted path across the converged replications); "mean_ci_width", "median_ci_width", "sd_ci_width" (summaries of the realized interval widths); "pct_ci_less_w" (proportion of realized widths at or below desired_width); "pct_ci_miss_low" and "pct_ci_miss_high" (tail-specific empirical non-coverage of the population path); "total_type_I_error" (overall empirical non-coverage, the sum of the two tails); and the echoes "suc_rep" (number of converged replications), "total_N" (the *N* evaluated), "true_path" (the population value of the targeted path under true_Sigma), "width", "conf_level", and "assurance" (present only when an assurance was supplied). The proportion rows are on the 0 to 1 scale, not percentages.

Note

Occasionally a replication fails to converge when the analysis model is fit to a simulated data set. Such replications are not counted toward the G converged replications; the simulation draws fresh data and continues. A safety cap stops the loop after 20 * G attempts, and a single warning is issued if fewer than G replications converged.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Lai, K., & Kelley, K. (2011). Accuracy in parameter estimation for targeted effects in structural equation modeling: Sample size planning for narrow confidence intervals. *Psychological Methods, 16*(2), 127–148. doi:10.1037/a0021764

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48*(2), 1–36. doi:10.18637/jss.v048.i02

See Also

sem, cov_sem, ss_aipe_sem_path

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# The planning values a researcher would bring to ss_aipe_sem_path(): two
# factors, each measured by three indicators, joined by a structural path
# of 0.5, with every residual variance at 0.5.
planning_model <- "
  f1 =~ 1*y1 + 0.8*y2 + 0.8*y3
  f2 =~ 1*y4 + 0.8*y5 + 0.8*y6
  f2 ~ 0.5*f1
  f1 ~~ 1*f1
  f2 ~~ 0.75*f2
  y1 ~~ 0.5*y1; y2 ~~ 0.5*y2; y3 ~~ 0.5*y3
  y4 ~~ 0.5*y4; y5 ~~ 0.5*y5; y6 ~~ 0.5*y6
"

# The population the study will actually sample from: the same structural
# path, but noisier indicators than the planning values assumed, with
# every residual variance at 0.8.
true_model <- "
  f1 =~ 1*y1 + 0.8*y2 + 0.8*y3
  f2 =~ 1*y4 + 0.8*y5 + 0.8*y6
  f2 ~ 0.5*f1
  f1 ~~ 1*f1
  f2 ~~ 0.75*f2
  y1 ~~ 0.8*y1; y2 ~~ 0.8*y2; y3 ~~ 0.8*y3
  y4 ~~ 0.8*y4; y5 ~~ 0.8*y5; y6 ~~ 0.8*y6
"

# The free analysis model, with the targeted path labeled "b".
analysis_model <- "
  f1 =~ y1 + y2 + y3
```

```
  f2 =~ y4 + y5 + y6
  f2 ~ b*f1
"

est_Sigma <- cov_sem(planning_model)$sigma_theta
true_Sigma <- cov_sem(true_model)$sigma_theta

# The sample size planned from the optimistic measurement quality is
# evaluated against the population that actually holds. Notice that the
# mean realized width (mean_ci_width) exceeds the desired 0.30, and that
# pct_ci_less_w, the proportion of intervals no wider than desired, falls
# well below the one half that the planned sample size would deliver if
# the planning values were right. G = 20 keeps the example quick; a
# reported sensitivity study deserves the default G = 100 or more.
set.seed(113)
ss_aipe_sem_path_sensitivity(model = analysis_model, est_Sigma = est_Sigma,
                             true_Sigma = true_Sigma, which_path = "b",
                             desired_width = 0.30, G = 20)
```

`ss_aipe_semipartial_r` *Sample Size for AIPE on a Semipartial (Part) Correlation*

Description

Determines the sample size needed for a confidence interval on a population semipartial correlation $r_{Y(X \cdot Z_1 \cdots Z_J)}$ (the unique contribution of X to Y after controlling for $Z_1, \ldots, Z_J$, with Y *not* residualized) to have a desired width, using the Olkin-Finn (1995) / Algina-Olejnik (2003) asymptotic variance and the AIPE framework of Kelley & Maxwell (2003).

Usage

```
ss_aipe_semipartial_r(
  r_sp,
  J,
  width,
  which_width = c("Full", "Lower", "Upper"),
  conf_level = 0.95,
  assurance = NULL
)
```

Arguments

`r_sp`	Anticipated population semipartial correlation, in $(-1, 1)$.
`J`	Number of variables partialled out of X (count of $Z_1, \ldots, Z_J$); must be at least 1.
`width`	Desired full width of the confidence interval on the semipartial correlation.

`which_width`	Whether `width` refers to the `"Full"` width (default) or the `"Lower"`/`"Upper"` half-width.
`conf_level`	Desired confidence level (default `0.95`).
`assurance`	Optional. Probability that the realized CI is no wider than `width`. When supplied, the sample size is inflated using the standard chi squared correction (Kelley & Maxwell, 2003).

Details

Asymptotic variance of the semipartial. Under multivariate normality, the sample semipartial correlation $r_{Y(X \cdot Z)}$ has asymptotic variance

$$\mathrm{Var}(\hat{r}_{Y(X \cdot Z)}) \approx \frac{(1 - r^2_{Y(X \cdot Z)})^2}{n - J - 1}$$

(Olkin & Finn, 1995, with the partial-correlation degrees-of-freedom correction). Inverting for the sample size needed to achieve a target half-width $w_{1/2}$ at confidence level $1 - \alpha$:

$$n = J + 1 + \left\lceil z^2_{1-\alpha/2} \cdot (1 - r^2_{Y(X \cdot Z)})^2 / w^2_{1/2} \right\rceil.$$

Comparison with partial-r planning. The partial correlation $r_{XY \cdot Z}$ divides the covariance after residualizing both X and Y on Z; the semipartial divides after residualizing only X. The semipartial is the natural effect size companion to a standardized regression coefficient: its square equals the ΔR^2 contributed by X above and beyond the controls. See `var_semipartial_r` for the asymptotic variance, and `ss_aipe_partial_r` for the partial-correlation analog of this function.

Note on conservatism of the assurance plan. The empirical simulation study of the AIPE planner family finds that `ss_aipe_semipartial_r()` is on the boundary of its valid range at 80% assurance and modestly conservative at 99% assurance. At $\gamma = 0.80$, the realized assurance at the recommended sample size is within Monte Carlo error of the target, that is, the bound is operating at the edge of its validity. At $\gamma = 0.99$, the ideal sample size is about 15 to 20 subjects smaller than the recommended sample size, reflecting the looser upper-tail bound at the 99% level. The recommended sample size is therefore a sufficient sample size rather than the smallest possible sample size. A small safety margin (5 to 10 subjects) is advisable when planning at $\gamma = 0.80$. `ss_aipe_semipartial_r_sensitivity` quantifies the overshoot for any one condition.

Value

A `data.frame` with rows for the recommended sample size, the expected CI width at that sample size, and the inputs echoed back.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Algina, J., & Olejnik, S. (2003). Sample size tables for correlation analysis with applications in partial correlation and multiple regression analysis. *Multivariate Behavioral Research, 38*(3), 309–323. doi:10.1207/s15327906mbr3803_02

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (3rd ed.). Lawrence Erlbaum.

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on the one-way ANOVA and Chapter 4 on contrasts.)

Olkin, I., & Finn, J. D. (1995). Correlations redux. *Psychological Bulletin, 118*(1), 155–164. doi:10.1037/00332909.118.1.155

See Also

`var_semipartial_r`, `ss_aipe_partial_r`, `ss_aipe_R2`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: `ss_aipe_c_sensitivity()`, `ss_aipe_cliff_delta()`, `ss_aipe_cliff_delta_sensit`
`ss_aipe_composite_sem()`, `ss_aipe_equivalence_r()`, `ss_aipe_equivalence_r_sensitivity()`,
`ss_aipe_equivalence_smd()`, `ss_aipe_equivalence_smd_sensitivity()`, `ss_aipe_icc()`, `ss_aipe_icc_sensitivi`
`ss_aipe_indirect_effect()`, `ss_aipe_indirect_effect_sensitivity()`, `ss_aipe_mixed_effects_sensitivity()`,
`ss_aipe_omega_squared()`, `ss_aipe_omega_squared_sensitivity()`, `ss_aipe_partial_r()`,
`ss_aipe_partial_r_sensitivity()`, `ss_aipe_pcm_sensitivity()`, `ss_aipe_r()`, `ss_aipe_r_sensitivity()`,
`ss_aipe_reliability_sensitivity()`, `ss_aipe_semipartial_r_sensitivity()`

Examples

```
# 1. Plan n so the 95% CI on r_sp (J = 3) has full width <= 0.15
#        when the anticipated semipartial is 0.25.
ss_aipe_semipartial_r(r_sp = 0.25, J = 3, width = 0.15)

# 2. With 80% assurance:
ss_aipe_semipartial_r(r_sp = 0.25, J = 3, width = 0.15,
                      assurance = 0.80)
```

`ss_aipe_semipartial_r_sensitivity`
Sensitivity Analysis for Sample Size Planning From the AIPE Perspective for a Semipartial Correlation

---

Description

Quantifies how much misspecification of the population semipartial correlation distorts an AIPE-based sample size plan. The function constructs a population covariance matrix whose implied semipartial correlation between Y and X_1 (partialing $X_2, \ldots, X_J$ out of X_1 only, not out of Y) equals `true_r_sp`, then on each replication draws an n-row sample, computes the sample semipartial correlation, and forms a Fisher's Z-style CI scaled by the standardized regression coefficient.

Usage

```
ss_aipe_semipartial_r_sensitivity(
  true_r_sp = NULL,
  estimated_r_sp = NULL,
  J,
  width,
  specified_N = NULL,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

`true_r_sp`	Population semipartial correlation; must lie in $(-1, 1)$.
`estimated_r_sp`	Planning value passed to `ss_aipe_semipartial_r`; supply this or `specified_N` but not both.
`J`	Total number of predictors. Must be at least 1.
`width`	Desired full width of the CI on the semipartial correlation.
`specified_N`	Sample size to evaluate.
`conf_level`	Confidence level (default `0.95`).
`assurance`	Optional assurance probability.
`G`	Number of Monte Carlo replications.
`print_iter`	Logical.
`filename`	Optional path for a comma separated file recording every replication (the sample semipartial correlation, the two confidence limits, the interval width, and two indicators of whether the interval missed `true_r_sp` below or above): nothing is written when `filename` is `NULL` (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at `tempfile(fileext = ".csv")`.

Value

A `data.frame` with rows for the realized semipartial correlation, the interval width, the proportion of intervals at or below `width`, tail-specific and overall non-coverage of `true_r_sp`, and the input echoes, including `assurance` (present only when an assurance was supplied).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

ss_aipe_semipartial_r, ss_aipe_partial_r_sensitivity

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other AIPE sample size planning: ss_aipe_c_sensitivity(), ss_aipe_cliff_delta(), ss_aipe_cliff_delta_sensit
ss_aipe_composite_sem(), ss_aipe_equivalence_r(), ss_aipe_equivalence_r_sensitivity(),
ss_aipe_equivalence_smd(), ss_aipe_equivalence_smd_sensitivity(), ss_aipe_icc(), ss_aipe_icc_sensitivi
ss_aipe_indirect_effect(), ss_aipe_indirect_effect_sensitivity(), ss_aipe_mixed_effects_sensitivity(),
ss_aipe_omega_squared(), ss_aipe_omega_squared_sensitivity(), ss_aipe_partial_r(),
ss_aipe_partial_r_sensitivity(), ss_aipe_pcm_sensitivity(), ss_aipe_r(), ss_aipe_r_sensitivity(),
ss_aipe_reliability_sensitivity(), ss_aipe_semipartial_r()

Examples

```
set.seed(113)
ss_aipe_semipartial_r_sensitivity(
  true_r_sp = 0.30, estimated_r_sp = 0.30, J = 3, width = 0.20,
  G = 50, print_iter = FALSE
)
```

---

ss_aipe_sm *Sample Size Planning for Accuracy in Parameter Estimation (AIPE) of the Standardized Mean*

Description

Plans the sample size needed for a sufficiently narrow confidence interval for the population standardized mean, the mean divided by the standard deviation, from the accuracy in parameter estimation (AIPE) perspective.

Usage

```
ss_aipe_sm(sm, width, conf_level = 0.95, assurance = NULL, ...)
```

Arguments

sm	The population standardized mean
width	The desired full width of the obtained confidence interval
conf_level	The desired confidence interval coverage, (i.e., 1 - Type I error rate)
assurance	Parameter to ensure that the obtained confidence interval width is narrower than the desired width with a specified degree of certainty (must be NULL or between zero and unity)
...	Allows one to potentially include parameter values for inner functions

Value

A 1-row data.frame with columns term and value:

necessary_N	The necessary total sample size in order to achieve the desired degree of accuracy (i.e., the sufficiently narrow confidence interval)

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*(4), 532–574. doi:10.1177/0013164401614002

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Kelley, K. (2005). The effects of nonnormal distributions on confidence intervals around the standardized mean difference: Bootstrap and parametric confidence intervals, *Educational and Psychological Measurement, 65*, 51–69. doi:10.1177/0013164404264850

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

ci_nc_t, ci_sm

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Suppose the population mean is believed to be 20, and the population
# standard deviation is believed to be 2; thus the population standardized
# mean is believed to be 10. To determine the necessary sample size for a
# study so that the full width of the 95 percent confidence interval
# obtained in the study will be, with 90% assurance, no wider than 2.5,
# the function should be specified as follows.

ss_aipe_sm(sm = 10, width = 2.5, conf_level = .95, assurance = .90)
```

---

ss_aipe_sm_sensitivity

Sensitivity Analysis for Sample Size Planning for the Standardized Mean From the Accuracy in Parameter Estimation (AIPE) Perspective

---

Description

Performs a sensitivity analysis when planning sample size from the Accuracy in Parameter Estimation (AIPE) Perspective for the standardized mean.

Usage

```
ss_aipe_sm_sensitivity(
  true_sm = NULL,
  estimated_sm = NULL,
  desired_width = NULL,
  specified_N = NULL,
  assurance = NULL,
  conf_level = 0.95,
  G = 10000,
  print_iter = TRUE,
  filename = NULL
)
```

Arguments

`true_sm`	population standardized mean
`estimated_sm`	estimated standardized mean
`desired_width`	desired full width of the confidence interval for the population standardized mean
`specified_N`	selected sample size to use in order to determine distributional properties of a given value of sample size
`assurance`	parameter to ensure that the obtained confidence interval width is narrower than the desired width with a specified degree of certainty (must be `NULL` or between zero and unity)

conf_level	the desired confidence interval coverage, (i.e., 1 - Type I error rate)
G	number of generations (i.e., replications) of the simulation
print_iter	to print the current value of the iterations
filename	Optional path to a CSV file; when supplied, the per-replication results (the observed standardized mean, the full and one-sided widths, the tail misses, and the interval limits) are written there, appended when the file already exists, and a throwaway run should point it at tempfile(fileext = ".csv"); the default NULL writes nothing.

Value

A data.frame with columns term and value summarizing the Monte Carlo sensitivity analysis across the G replications. The term entries are: mean_sm, median_sm, sd_sm (summaries of the realized standardized mean); mean_ci_width, median_ci_width, sd_ci_width (summaries of the full interval widths); mean_ci_width_lower and mean_ci_width_upper (mean one-sided widths, measured from the observed standardized mean to each limit); pct_ci_less_w (proportion of intervals at or below the target width); pct_ci_miss_low and pct_ci_miss_high (tail-specific empirical non-coverage of true_sm); total_type_I_error (overall empirical non-coverage, the sum of the two tails); and the input echoes total_N, true_sm, estimated_sm (NA when specified_N was supplied instead), width, conf_level, and assurance (present only when an assurance was supplied). The proportion and Type I error rows are proportions on the 0 to 1 scale, not percentages.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*(4), 532–574. doi:10.1177/0013164401614002

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Kelley, K. (2005). The effects of nonnormal distributions on confidence intervals around the standardized mean difference: Bootstrap and parametric confidence intervals, *Educational and Psychological Measurement, 65*, 51–69. doi:10.1177/0013164404264850

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

ss_aipe_sm

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# With true_sm equal to estimated_sm the planning value is correct, and
# the sweep reports what a correctly specified plan delivers: the share of
# intervals no wider than the target should sit near the assurance. G = 10
# keeps the example quick; a reported sensitivity study deserves the
# default G = 10000.
set.seed(113)
res_correct <- ss_aipe_sm_sensitivity(true_sm = 1.5, estimated_sm = 1.5,
  desired_width = 0.5, assurance = 0.95, conf_level = 0.95, G = 10,
  print_iter = FALSE)
res_correct

# The terms the summary reports.
res_correct$term

# The proportion of realized full widths no wider than the target.
res_correct$value[res_correct$term == "pct_ci_less_w"]

# The population standardized mean is 1.5 but the plan assumed 2, so the
# planner sizes the study for a wider sampling distribution than the data
# will show, and the realized intervals come in narrower than the target.
set.seed(113)
res_misspecified <- ss_aipe_sm_sensitivity(true_sm = 1.5, estimated_sm = 2,
  desired_width = 0.5, G = 20, print_iter = FALSE)

# The effect of the misspecification on the mean interval width.
res_misspecified$value[res_misspecified$term == "mean_ci_width"]
```

---

ss_aipe_smd *Sample Size Planning for the Standardized Mean Difference (AIPE)*

Description

Determines the per-group sample size needed for a two-independent-groups design so that the (expected) confidence interval for Cohen's d, the population standardized mean difference, denoted δ, is no wider than a user-specified value. This is the Accuracy in Parameter Estimation (AIPE) framework of Kelley and Rausch (2006), the standardized-mean-difference companion to power-based planning via ss_power_smd. Optionally, supplying assurance returns the larger sample size needed so that the realized interval will be at or below the target width with that probability rather than just on average.

Usage

```
ss_aipe_smd(delta, conf_level = 0.95, width, assurance = NULL)
```

Arguments

`delta`	The supposed value of the population standardized mean difference δ the sample size is planned against: a value the researcher posits, either a minimally important effect or a value believed to be true in the population, never a sample estimate. Echoed in the returned table as the `supposed_smd` row.
`conf_level`	Desired confidence level (i.e., $1 - \alpha$, where α is the Type I error rate). Default `0.95`.
`width`	Desired (full) width of the two-sided confidence interval on δ.
`assurance`	Optional probability with which the realized confidence interval is to be no wider than `width`. When `NULL` (the default), the planning targets the *expected* width; when supplied (e.g., 0.80, 0.90, 0.99), the procedure returns the larger N that guarantees the desired width with that assurance. Must be `NULL` or strictly between 0.50 and 1.

Value

A `data.frame` with columns `term` and `value`. The first row, `necessary_n_per_group`, is the necessary per-group sample size N; the remaining rows echo the user-supplied planning inputs `supposed_smd` and `width` (and `assurance` when supplied), so the assumptions the sample size was planned under travel with the result. The `supposed_smd` row is the supposed effect the plan is built on: a value the researcher posits, either a minimally important effect or a value believed to be true in the population, never a sample estimate. The confidence level is reported in the printed footer.

Warning

The returned value is the sample size *per group*.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Anderson, S. F., & Kelley, K. (2024). Sample size planning for replication studies: The devil is in the design. *Psychological Methods, 29*(5), 844–867. doi:10.1037/met0000520

Anderson, S. F., Kelley, K., & Maxwell, S. E. (2017). Sample-size planning for more accurate statistical power: A method adjusting sample effect sizes for publication bias and uncertainty. *Psychological Science, 28*(11), 1547–1562. doi:10.1177/0956797617723724

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*(4), 532–574. doi:10.1177/0013164401614002

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Kelley, K. (2005). The effects of nonnormal distributions on confidence intervals around the standardized mean difference: Bootstrap and parametric confidence intervals, *Educational and Psychological Measurement, 65*, 51–69. doi:10.1177/0013164404264850

Kelley, K., Maxwell, S. E., & Rausch, J. R. (2003). Obtaining power or obtaining precision: Delineating methods of sample size planning. *Evaluation and the Health Professions, 26*(3), 258–287. doi:10.1177/0163278703255242

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons and Chapter 3 on one-way ANOVA.)

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

`smd`, `smd_c`, `ci_smd`, `ci_smd_c`, `ci_nc_t`, `stats::power.t.test()`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
ss_aipe_smd(delta = .5, conf_level = .95, width = .30)
ss_aipe_smd(delta = .5, conf_level = .95, width = .30, assurance = .8)
ss_aipe_smd(delta = .5, conf_level = .95, width = .30, assurance = .95)
```

`ss_aipe_smd_sensitivity`

Sensitivity Analysis for Sample Size Given the Accuracy in Parameter Estimation Approach for the Standardized Mean Difference

---

Description

Performs sensitivity analysis for sample size determination for the standardized mean difference given a population and a standardized mean difference. Allows one to determine the effect of being wrong when estimating the population standardized mean difference in terms of the width of the obtained (two-sided) confidence intervals.

Usage

```
ss_aipe_smd_sensitivity(
  true_delta = NULL,
  estimated_delta = NULL,
  desired_width = NULL,
  n_per_group = NULL,
  assurance = NULL,
  conf_level = 0.95,
  G = 1000,
  print_iter = FALSE,
  filename = NULL
)
```

Arguments

true_delta	population standardized mean difference
estimated_delta	estimated standardized mean difference; can be true_delta to perform standard simulations
desired_width	describe full width for the confidence interval around the population standardized mean difference
n_per_group	selected sample size to use in order to determine distributional properties of at a given value of sample size
assurance	parameter to ensure confidence interval width with a specified degree of certainty (must be NULL or between zero and unity)
conf_level	the desired degree of confidence (i.e., 1-Type I error rate)
G	number of generations (i.e., replications) of the simulation
print_iter	to print the current value of the iterations
filename	an optional path for a comma separated file recording every replication (the realized standardized mean difference, the full and the two one-sided interval widths, the three non-coverage indicators, and the two confidence limits): nothing is written when filename is NULL (the default), a new file with a header row is created otherwise, an existing file at that path is appended to, and a throwaway run should point it at tempfile(fileext = ".csv")

Details

For sensitivity analysis when planning sample size given the desire to obtain narrow confidence intervals for the population standardized mean difference. Given a population value and an estimated value, one can determine the effects of incorrectly specifying the population standardized mean difference (true_delta) on the obtained widths of the confidence intervals. Also, one can evaluate the percent of the confidence intervals that are less than the desired width (especially when modifying the assurance parameter); see ss_aipe_smd) Alternatively, one can specify n_per_group to determine the results at a particular sample size (when doing this estimated_delta cannot be specified).

Value

A `data.frame` with columns `term` and `value` summarizing the Monte Carlo sensitivity analysis across the `G` replications. The `term` entries are: `mean_smd`, `median_smd`, `sd_smd` (summaries of the realized standardized mean difference); `mean_ci_width`, `median_ci_width`, `sd_ci_width` (summaries of the full interval widths); `mean_ci_width_lower` and `mean_ci_width_upper` (mean one-sided widths, measured from the observed standardized mean difference to each limit); `pct_ci_less_w` (proportion of intervals at or below the target width); `pct_ci_miss_low` and `pct_ci_miss_high` (tail-specific empirical non-coverage of `true_delta`); `total_type_I_error` (overall empirical non-coverage, the sum of the two tails); and the input echoes `n_per_group`, `total_N`, `true_delta`, `estimated_delta` (NA when `n_per_group` was supplied instead), `width`, `conf_level`, and `assurance` (present only when an assurance was supplied). The proportion and Type I error rows are proportions on the 0 to 1 scale, not percentages.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement, 61*(4), 532–574. doi:10.1177/0013164401614002

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Kelley, K. (2005). The effects of nonnormal distributions on confidence intervals around the standardized mean difference: Bootstrap and parametric confidence intervals, *Educational and Psychological Measurement, 65*, 51–69. doi:10.1177/0013164404264850

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons and Chapter 3 on one-way ANOVA.)

Steiger, J. H., & Fouladi, R. T. (1997). Noncentrality interval estimation and the evaluation of statistical methods. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

`ss_aipe_smd`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Since 'true_delta' equals 'estimated_delta', this usage
# returns the results of a correctly specified situation.
```

```
# Note that 'G' should be large (50 is used to make the example run easily)
set.seed(113)
Res.1 <- ss_aipe_smd_sensitivity(true_delta=.5, estimated_delta=.5, desired_width=.30,
                                 assurance=NULL, conf_level=.95, G=50, print_iter=FALSE)

# Objects contained in the 'summary'.
Res.1$term

# True standardized mean difference is .4, but specified at .5.
# Change 'G' to some large number (e.g., G=5,000)
Res.2 <- ss_aipe_smd_sensitivity(true_delta=.4, estimated_delta=.5, desired_width=.30,
                                 assurance=NULL, conf_level=.95, G=50, print_iter=FALSE)

# The effect of the misspecification on mean confidence intervals is:
Res.2[1,]

# True standardized mean difference is .5, but specified at .4.
Res.3 <- ss_aipe_smd_sensitivity(true_delta=.5, estimated_delta=.4, desired_width=.30,
                                 assurance=NULL, conf_level=.95, G=50, print_iter=FALSE)

# The effect of the misspecification on mean confidence intervals is:
Res.3[1,]
```

`ss_aipe_src` *Sample Size Necessary for the Accuracy in Parameter Estimation Approach for a Standardized Regression Coefficient of Interest*

Description

A function used to plan sample size from the accuracy in parameter estimation approach for a standardized regression coefficient of interest given the input specification.

Usage

```
ss_aipe_src(
  rho2_Y_X = NULL,
  Rho2_j_X_without_j = NULL,
  p = NULL,
  beta_j = NULL,
  width,
  which_width = "Full",
  sigma_Y = 1,
  sigma_X_j = 1,
  rho_XX = NULL,
  rho_YX = NULL,
  which_predictor = NULL,
  alpha_lower = NULL,
  alpha_upper = NULL,
```

```
  conf_level = 0.95,
  assurance = NULL
)
```

Arguments

rho2_Y_X	Population value of the squared multiple correlation coefficient
Rho2_j_X_without_j	Population value of the squared multiple correlation coefficient predicting the *j*th predictor variable from the remaining *p*-1 predictor variables
p	The number of predictor variables
beta_j	The regression coefficient for the *j*th predictor variable (i.e., the predictor of interest)
width	The desired width of the confidence interval
which_width	Which width ("Full", "Lower", or "Upper") the width refers to (at present, only "Full" can be specified)
sigma_Y	The population standard deviation of *Y* (i.e., the dependent variables)
sigma_X_j	The population standard deviation of the *j*th *X* variable (i.e., the predictor variable of interest)
rho_XX	Population correlation matrix for the *p* predictor variables
rho_YX	Population *p* length vector of correlation between the dependent variable (*Y*) and the *p* independent variables
which_predictor	Identifies which of the *p* predictors is of interest
alpha_lower	Type I error rate for the lower confidence interval limit
alpha_upper	Type I error rate for the upper confidence interval limit
conf_level	Desired level of confidence for the computed interval (i.e., 1 - the Type I error rate)
assurance	Degree of certainty that the obtained confidence interval will be sufficiently narrow, which yields an approximate sample size to be verified with function ss_aipe_reg_coef_sensitivity to determine if it is appropriate

Details

Not all of the arguments need to be specified, only those that provide all of the necessary information so that the sample size can be determined for the conditions specified.

Value

Returns the necessary sample size in order for the goals of accuracy in parameter estimation to be satisfied for the confidence interval for a particular regression coefficient given the input specifications.

Warning

As discussed in Kelley and Maxwell (2008), the sample size planning approach from the AIPE perspective used in this function is only an approximation.

Note

This function calls upon ss_aipe_reg_coef in DMAR but has a different naming scheme. See ss_aipe_reg_coef for more details.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means and Chapter 6 on trend analysis.)

See Also

ss_aipe_reg_coef_sensitivity, ci_nc_t, ss_aipe_reg_coef, ss_aipe_rc

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Exchangable correlation structure
rho_YX <- c(.3, .3, .3, .3, .3)
rho_XX <- rbind(c(1, .5, .5, .5, .5), c(.5, 1, .5, .5, .5), c(.5, .5, 1, .5, .5),
                c(.5, .5, .5, 1, .5), c(.5, .5, .5, .5, 1))

ss_aipe_src(width = .1, which_width = "Full", sigma_Y = 1, sigma_X_j = 1, rho_XX = rho_XX,
            rho_YX = rho_YX, which_predictor = 1, conf_level = 1 - .05)

ss_aipe_src(width = .1, which_width = "Full", sigma_Y = 1, sigma_X_j = 1, rho_XX = rho_XX,
            rho_YX = rho_YX, which_predictor = 1, conf_level = 1 - .05,
            assurance = .85)
```

ss_aipe_src_sensitivity

Sensitivity Analysis for Sample Size Planning From the Accuracy in Parameter Estimation Perspective for the Standardized Regression Coefficient

Description

Performs a sensitivity analysis when planning sample size from the Accuracy in Parameter Estimation Perspective for the standardized regression coefficient.

Usage

```
ss_aipe_src_sensitivity(
  true_var_Y = NULL,
  true_cov_YX = NULL,
  true_cov_XX = NULL,
  estimated_var_Y = NULL,
  estimated_cov_YX = NULL,
  estimated_cov_XX = NULL,
  specified_N = NULL,
  which_predictor = 1,
  w = NULL,
  noncentral = TRUE,
  standardize = TRUE,
  conf_level = 0.95,
  assurance = NULL,
  G = 1000,
  print_iter = TRUE,
  filename = NULL
)
```

Arguments

true_var_Y	Population variance of the dependent variable (Y)
true_cov_YX	Population covariances vector between the p predictor variables and the dependent variable (Y)
true_cov_XX	Population covariance matrix of the p predictor variables
estimated_var_Y	Estimated variance of the dependent variable (Y)
estimated_cov_YX	Estimated covariances vector between the p predictor variables and the dependent variable (Y)
estimated_cov_XX	Estimated Population covariance matrix of the p predictor variables

specified_N	Directly specified sample size (instead of planning one from the estimated covariance structure)
which_predictor	identifies which of the p predictors is of interest
w	desired confidence interval width for the regression coefficient of interest
noncentral	specify with a TRUE or FALSE statement whether or not the noncentral approach to sample size planning should be used
standardize	specify with a TRUE or FALSE statement whether or not the regression coefficient will be standardized; default is TRUE
conf_level	desired level of confidence for the computed interval (i.e., 1 - the Type I error rate)
assurance	degree of certainty that the obtained confidence interval will be sufficiently narrow
G	the number of generations/replication of the simulation study within the function
print_iter	specify with a TRUE/FALSE statement if the iteration number should be printed as the simulation within the function runs
filename	Optional path for a comma separated file recording every replication, forwarded to ss_aipe_reg_coef_sensitivity, which does the writing: nothing is written when filename is NULL (the default), and a throwaway run should point it at tempfile(fileext = ".csv").

Details

Direct specification of true_cov_YX and true_cov_XX is necessary, even if one is interested in a single regression coefficient, so that the covariance/correlation structure can be specified when the simulation study within the function runs.

Value

A data.frame with columns term and value summarizing the Monte Carlo sensitivity analysis. This function delegates to ss_aipe_reg_coef_sensitivity and inherits its return structure: mean / median / SD summaries of the realized standardized regression coefficient, the realized interval widths, and the realized squared multiple correlation coefficient; the proportion of intervals at or below the planning target (pct_ci_less_w); the tail-specific and overall empirical non-coverage rates (pct_ci_miss_low, pct_ci_miss_high, total_type_I_error), all proportions on the 0 to 1 scale; and the input echoes (total_N, p, which_predictor, true_b_j, estimated_b_j, width, conf_level, and, when one was supplied, assurance). See ss_aipe_reg_coef_sensitivity for the full row list.

Note

Note that when the true and estimated covariance structures agree (true_cov_YX equals estimated_cov_YX and true_cov_XX equals estimated_cov_XX), the results are not literally from a sensitivity analysis, rather the function performs a standard simulation study. A simulation study can be helpful in order to determine if the sample size procedure under or overestimates necessary sample size. See ss_aipe_reg_coef_sensitivity in DMAR for more details.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means and Chapter 6 on trend analysis.)

See Also

ss_aipe_reg_coef_sensitivity, ss_aipe_rc_sensitivity, ss_aipe_reg_coef, ci_reg_coef

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Examples

```
# Sensitivity analysis for a standardized regression coefficient
# with two correlated predictors. A production run uses many more
# generations (G = 1000 is typical); G is reduced here so the
# example runs quickly.
set.seed(113)
Sigma_X <- matrix(c(1, 0.3, 0.3, 1), nrow = 2)
cov_YX  <- c(0.4, 0.3)
ss_aipe_src_sensitivity(
  true_var_Y = 1, true_cov_YX = cov_YX, true_cov_XX = Sigma_X,
  estimated_var_Y = 1, estimated_cov_YX = cov_YX, estimated_cov_XX = Sigma_X,
  which_predictor = 1, w = 0.20, conf_level = 0.95,
  G = 50, print_iter = FALSE
)
```

---

ss_power_c — *Sample Size or Power for an Unstandardized Contrast in a One-Way Between-Subjects ANOVA*

Description

Determine the necessary per-group sample size to achieve a desired level of statistical power for the test of a single planned (unstandardized) contrast in a one-way between-subjects analysis of variance, or, given a per-group sample size, return the realized statistical power.

Usage

```
ss_power_c(
  psi,
  c_weights,
  sigma,
  desired_power = 0.85,
  alpha_level = 0.05,
  n = NULL,
  directional = FALSE
)
```

Arguments

psi	The population unstandardized contrast effect, $\psi = \sum c_j \mu_j$
c_weights	Vector of contrast weights (must sum to zero); use fractional weights so that the positive weights sum to 1 (e.g., c(0.5, 0.5, -0.5, -0.5))
sigma	Within-group population standard deviation
desired_power	Desired statistical power (default 0.85)
alpha_level	Type I error rate (default 0.05)
n	Per-group sample size (assumed balanced); if specified, returns the realized power
directional	Logical: TRUE for a one-sided test (in the same sign as psi), FALSE (default) for a two-sided test

Details

Under the alternative hypothesis the contrast *t*-statistic follows a noncentral *t*-distribution with degrees of freedom $N - J$ (where $N = nJ$ is the total sample size and J the number of groups, taken as length(c_weights)) and noncentrality parameter $\lambda = \psi/(\sigma\sqrt{\sum c_j^2/n})$.

The function searches over per-group sample sizes n until power first reaches desired_power; when n is supplied it instead returns the realized power.

Value

A data.frame with rows for necessary_n_per_group (or specified_n_per_group), actual_power, and noncentral_t_parm. The result carries the dmar_ss_power class, so tidy and glance summarize it in broom convention.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

ss_power_sc, ss_power_one_way_anova, ss_power_c_ancova, ci_c, ss_aipe_c

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_ ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova ss_power_composite_factorial_anova(), ss_power_composite_sem(), ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# Power for the contrast (Group 1 + Group 2) / 2 vs (Group 3 + Group 4) / 2
# with population contrast = 0.5, within-group sigma = 1, desired power = .80
ss_power_c(psi = 0.5, c_weights = c(0.5, 0.5, -0.5, -0.5), sigma = 1,
           desired_power = 0.80)

# Realized power for n = 30 per group
ss_power_c(psi = 0.5, c_weights = c(0.5, 0.5, -0.5, -0.5), sigma = 1, n = 30)
```

---

ss_power_c_ancova — *Sample Size or Power for an Unstandardized Contrast in a One-Way ANCOVA*

Description

Determine the necessary per-group sample size to achieve a desired level of statistical power for the test of a single planned (unstandardized) contrast on the adjusted means in a one-way analysis of covariance, or, given a per-group sample size, return the realized statistical power.

Usage

```
ss_power_c_ancova(
  psi,
  c_weights,
  sigma,
  rho,
  desired_power = 0.85,
  alpha_level = 0.05,
  n = NULL,
  directional = FALSE
)
```

Arguments

psi	The population unstandardized contrast effect on the adjusted means, $\psi = \sum c_j \mu_j^{(adj)}$
c_weights	Vector of contrast weights (must sum to zero); use fractional weights so the positive weights sum to 1
sigma	Within-group population standard deviation of the response (the same σ as in a one-way ANOVA on the response)
rho	Within-group population correlation between the response and the covariate; must lie in (-1, 1)
desired_power	Desired statistical power (default 0.85)
alpha_level	Type I error rate (default 0.05)
n	Per-group sample size (assumed balanced); if specified, returns the realized power
directional	Logical: TRUE for a one-sided test (in the same sign as psi), FALSE (default) for a two-sided test

Details

This function uses the standard large-sample formulation in which the ANCOVA error variance is $\sigma^2_{adj} = \sigma^2(1 - \rho^2)$, the contrast *t*-statistic has degrees of freedom $N - J - 1$ (one less than the corresponding ANOVA contrast because of the covariate), and the noncentrality parameter is $\lambda = \psi/(\sigma\sqrt{1-\rho^2}\sqrt{\sum c_j^2/n})$. This assumes the covariate means are equal across groups (the typical assumption under random assignment); for designs with substantial group differences in the covariate, the small-sample correction $1 + (\bar{X}_j - \bar{X}_\cdot)^2/SS_X^{(within)}$ would slightly inflate the standard error and reduce power, an effect that is negligible for moderate or large n.

Value

A data.frame with rows for necessary_n_per_group (or specified_n_per_group), actual_power, and noncentral_t_parm. The result carries the dmar_ss_power class, so tidy and glance summarize it in broom convention.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

See Also

ss_power_c, ci_c_ancova, ss_aipe_c_ancova

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova ss_power_composite_factorial_anova(), ss_power_composite_sem(), ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# Same population contrast as in the ANOVA example, with rho = 0.5 between
# outcome and covariate; ANCOVA is more efficient than ANOVA here.
ss_power_c_ancova(psi = 0.5, c_weights = c(0.5, 0.5, -0.5, -0.5),
                  sigma = 1, rho = 0.5, desired_power = 0.80)

# Realized power for n = 30 per group
ss_power_c_ancova(psi = 0.5, c_weights = c(0.5, 0.5, -0.5, -0.5),
                  sigma = 1, rho = 0.5, n = 30)
```

---

ss_power_composite_ancova

Sample Size or Composite Power for a One-Way or Factorial ANCOVA

Description

Determine the necessary per-cell sample size to achieve a desired level of composite statistical power in a balanced analysis of covariance with a groups (a one-way design) or a factorial arrangement of factors, or, given a per-cell sample size, return the realized composite power. Composite power is the probability that every effect named in `effects` is statistically significant in the same study, the quantity a design must be planned against when its conclusion requires more than one result to hold at once.

Usage

```
ss_power_composite_ancova(
  factor_levels,
  effects,
  slopes = c("homogeneous", "heterogeneous"),
  means = NULL,
  sigma = NULL,
  covariate_R2 = 0,
  n_covariates = 0,
  correlations = NULL,
  sd_cov = 1,
  desired_power = 0.85,
  alpha_level = 0.05,
  n_per_cell = NULL
)
```

Arguments

`factor_levels`	Integer vector of the number of levels of each factor, one entry per factor (each at least 2). A single value a is a one-way design with a groups; `c(2, 3)` is a 2 by 3 factorial.
`effects`	A non-empty list naming the effects in the composite. For `slopes = "homogeneous"` each element has `factors` (the factor indices the effect spans) and, unless `means` is supplied, its effect size `f` or `partial_eta_squared`. For `slopes = "heterogeneous"` each element also has a `type`, one of `"mean"` (a factorial mean effect, the default), `"covariate"` (the average slope, spanning no factors), or `"slope"` (a factor-by-covariate slope heterogeneity). An optional `label` names the effect. See the forwarded functions for the exact grammar.
`slopes`	The covariate-slope model, `"homogeneous"` (one common slope, the default) or `"heterogeneous"` (the slope may differ across cells, making the covariate and slope-heterogeneity effects testable).
`means`	Optional array of population cell means (dimensions `factor_levels`, or a vector in array order) from which the mean effects' sizes are read; enables the mean-pattern figure.
`sigma`	The common within-cell population standard deviation of the outcome, required with the population-values interface.
`covariate_R2`	For `slopes = "homogeneous"` only: proportion of the outcome's within-cell variance the covariate or covariates explain, in $[0, 1)$, which raises every effect's noncentrality through $f/\sqrt{1 - R^2}$. Defaults to 0.
`n_covariates`	For `slopes = "homogeneous"` only: number of covariates, each spending one residual degree of freedom. Must be positive when `covariate_R2` is. Defaults to 0.
`correlations`	For `slopes = "heterogeneous"` only: optional array of the population covariate-outcome correlation within each cell (dimensions `factor_levels`), which supplies the covariate and slope effects' sizes with the population-values interface.

sd_cov	For slopes = "heterogeneous" only: population standard deviation of the covariate, the units the slopes and the figure are drawn in. Scale free, so it changes no power. Default 1.
desired_power	Desired composite statistical power (default 0.85). Used only when n_per_cell is NULL.
alpha_level	Type I error rate for each individual F test (default 0.05), the per-test rate, not a rate for the composite event.
n_per_cell	Per-cell sample size (balanced); if supplied, the realized composite power is returned rather than a sample size planned.

Details

This is the general entry point for the ANCOVA composite. The two-group design has its own simpler interface in ss_power_composite_ancova_2group, stated through a single smd and one or two correlations; use that when there are exactly two groups. Use this function for more than two groups or for a factorial design. With no covariate, name the effects through ss_power_composite_anova instead.

Homogeneous or heterogeneous slopes. The slopes argument chooses the model. "homogeneous" (the default) assumes one common covariate slope across the cells; the covariate is a variance reducer and the composite is over the factorial mean effects. "heterogeneous" lets the slope differ across cells, which makes the average covariate slope and the factor-by-covariate slope heterogeneity into testable effects that can join the mean effects in the composite. The heterogeneous one-way case is the a-group generalization of the two-group ANCOVA composite: a group mean effect, the covariate effect, and the group-by-covariate slope heterogeneity.

This function is a thin dispatcher. slopes = "homogeneous" forwards to ss_power_composite_factorial_ancova and slopes = "heterogeneous" to ss_power_composite_factorial_ancova_het; see those for the full method, the exactness discussion, and the shape of the returned object. The population effects can be stated as effect sizes or as population values (cell means, and for heterogeneous slopes a covariate-outcome correlation per cell) with a common within-cell standard deviation, from which the plot() method draws the population pattern.

Value

The data.frame the forwarded planner returns, with term and value columns and the dmar_ss_power class for tidy and glance, plus the dmar_composite_power_factorial (homogeneous) or dmar_composite_power_facto (heterogeneous) class for plot(). See ss_power_composite_factorial_ancova for the row-by-row description.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E. (2004). The persistence of underpowered studies in psychological research: Causes, consequences, and remedies. *Psychological Methods, 9*, 147–163.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9 on the analysis of covariance, Chapter 7 on factorial designs, and Chapter 3 on statistical power.)

See Also

ss_power_composite_ancova_2group for the two-group special case with the simple smd/rho interface; ss_power_composite_anova for the no-covariate design; ss_power_composite_factorial_ancova and ss_power_composite_factorial_ancova_het, the planners this forwards to; ss_power_factorial_ancova for a single effect

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova, ss_power_composite_factorial_anova(), ss_power_composite_sem(), ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Other composite power: ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_sem()

Examples

```
# A four-group (one-way) ANCOVA with heterogeneous slopes: the a-group
# generalization of the two-group composite. The conclusion needs the group
# mean effect, a covariate effect, and evidence that the covariate slope
# differs across the groups, so the design is planned against all three.
ss_power_composite_ancova(
  factor_levels = 4, slopes = "heterogeneous",
  effects = list(list(type = "mean",      factors = 1, f = 0.30),
                 list(type = "covariate",              f = 0.40),
                 list(type = "slope",      factors = 1, f = 0.20)),
  desired_power = 0.80)

# A 2 by 3 factorial ANCOVA with one common slope. Both main effects must
# hold; the covariate explains 25 percent of the within-cell variance.
ss_power_composite_ancova(
  factor_levels = c(2, 3),
  effects = list(list(factors = 1, f = 0.25),
                 list(factors = 2, f = 0.20)),
  covariate_R2 = 0.25, n_covariates = 1,
  desired_power = 0.80)

# The population effects can instead be a full pattern of cell means with a
# common within-cell SD; plot() then draws the mean pattern itself.
cell_means <- matrix(c(10, 12, 11,
                       13, 12, 16), nrow = 2, byrow = TRUE)
fit <- ss_power_composite_ancova(
  factor_levels = c(2, 3), means = cell_means, sigma = 4,
  effects = list(list(factors = 1, label = "A"),
```

```
                list(factors = 2, label = "B")),
  n_per_cell = 30)
fit
plot(fit)

# A one-row broom summary of a plan.
generics::tidy(ss_power_composite_ancova(
  factor_levels = c(2, 3),
  effects = list(list(factors = 1, f = 0.25),
                list(factors = 2, f = 0.20)),
  desired_power = 0.80))

# The two-group case matches the dedicated two-group planner. Here factor 1
# has two levels, so the heterogeneous one-way composite reproduces
# ss_power_composite_ancova_2group with one correlation per group.
ss_power_composite_ancova(
  factor_levels = 2, slopes = "heterogeneous",
  means = c(0, 0.5), correlations = c(0.1, 0.5), sigma = 1,
  effects = list(list(type = "mean", factors = 1),
                list(type = "covariate"),
                list(type = "slope", factors = 1)),
  n_per_cell = 95)
ss_power_composite_ancova_2group(smd = 0.5, rho = c(0.1, 0.5), n = 95)
```

`ss_power_composite_ancova_2group` *Sample Size or Composite Power for a Two-Group ANCOVA With a Covariate*

Description

Determine the necessary per-group sample size to achieve a desired level of composite statistical power in a two-group analysis of covariance, or, given a per-group sample size, return the realized composite power. Composite power is the probability that every effect named in `composite_terms` is statistically significant in the same study, which is the quantity a design has to be planned against when its conclusion requires more than one result to hold at once.

Usage

```
ss_power_composite_ancova_2group(
  smd = 0,
  rho = 0,
  sigma = 1,
  sd_cov = 1,
  composite_terms = c("group", "covariate", "group_by_covariate"),
  include_interaction = TRUE,
  desired_power = 0.85,
  alpha_level = 0.05,
```

```
  n = NULL,
  directional = FALSE
)

## S3 method for class 'dmar_composite_power'
plot(x, ...)
```

Arguments

smd	Supposed standardized mean difference (Cohen's *d*) between the two groups at the mean of the covariate, standardized by sigma: a value the researcher posits for the population, either a minimally important effect or a value believed to be true, never a sample estimate. Defaults to 0, which is no group effect and therefore power equal to alpha_level for that test.
rho	Supposed within-group population correlation between the covariate and the outcome. Length 1 for the same correlation in both groups, in which case the two slopes are equal and the interaction is zero, or length 2 for one correlation per group, in which case the slopes differ and the interaction carries the difference. Each element must lie in (-1, 1). Defaults to 0.
sigma	Within-group population standard deviation of the outcome: the same σ as in a one-way ANOVA on the outcome, the same in both groups, and the standardizer of smd. Defaults to 1, which puts smd and the coefficients on a standardized scale.
sd_cov	Population standard deviation of the covariate. Defaults to 1. A correlation is scale free, so sd_cov does not affect power; it sets the units the slopes and the figure are expressed in.
composite_terms	Character vector naming the effects that must all be statistically significant. Any subset of "group", "covariate", and "group_by_covariate". A single term returns that test's ordinary power. Defaults to all three.
include_interaction	Logical: whether the fitted model contains the group by covariate interaction. Defaults to TRUE. Dropping it returns one residual degree of freedom.
desired_power	Desired composite statistical power (default 0.85). Used only when n is NULL.
alpha_level	Type I error rate for each individual test (default 0.05). This is the per-test rate, not a rate for the composite event.
n	Per-group sample size (assumed balanced); if specified, the realized power is returned rather than a sample size planned.
directional	Logical: TRUE for one-sided tests, each in the direction of its own supposed effect, FALSE (default) for two-sided tests.
x	An object returned by ss_power_composite_ancova_2group.
...	Further arguments passed to the figure: cov_range, palette, group_labels, and show_power.

Details

This is the two-group special case, kept under its own name for the simple `smd`/`rho` interface it allows. For a one-way design with more than two groups, or a factorial design, use the general `ss_power_composite_ancova`.

The model is

$$Y = b_0 + b_{group}G + b_{cov}X + b_{group\times cov}GX + e.$$

The group effect, the covariate effect, and the group by covariate interaction can each be named in the composite, alone or in any combination.

Calling `plot` on the result draws the population effects the planning values describe. The figure needs only the effect sizes, and nothing is simulated to draw it.

Coding the group factor as -1/2 and +1/2 and centering the covariate makes each coefficient read directly: b_{group} is the difference between the group means at the covariate mean, b_{cov} is the average of the two within-group slopes, and $b_{group\times cov}$ is the difference between them. With balanced groups and a covariate whose distribution does not differ across them, the columns of the design are mutually orthogonal, so the three tests are orthogonal and the coefficient estimates are uncorrelated.

Orthogonal effects do not give independent tests. Every test divides by the same estimated error standard deviation, so an error estimate that lands low inflates all of the test statistics together. The tests are positively dependent, and composite power is strictly larger than the product of the marginal powers. Multiplying the marginal powers understates the composite; the gap closes as the residual degrees of freedom grow and the error estimate stabilizes. Composite power can never exceed the least powerful test in the set, so the weakest effect governs the design.

Conditional on the error estimate the tests are independent, which reduces the composite to a one-dimensional integral over the chi square distribution of that estimate. Adaptive quadrature evaluates the integral, so no data are simulated and the result is deterministic to quadrature precision.

Within group g the slope is $\rho_g\sigma/\sigma_X$ and the residual variance is $\sigma^2(1-\rho_g^2)$, so correlations that differ across the groups give slopes that differ. The error variance the ANCOVA pools and estimates is the average of the two, $\sigma^2_{adj} = \sigma^2(1-\bar{\rho^2})$ with $\bar{\rho^2}$ the mean of ρ_1^2 and ρ_2^2, which is the familiar $\sigma\sqrt{1-\rho^2}$ whenever the two correlations are equal in absolute value. Correlations that are not make the residual variance differ across the groups, and that has consequences the section on unequal residual variances below spells out.

The noncentralities are formed as $\sqrt{N}f$, the convention the rest of the `ss_power_*` family uses (see `ss_power_reg_coef`), and the residual degrees of freedom are $N-p-1$. When the correlations are equal and the interaction is dropped, the group test reproduces `ss_power_c_ancova` with contrast weights `c(1, -1)`.

Two approximations are worth separating, because they have different causes. The first is the conditioning on the covariate, which is present at every set of planning values and is described next. The second appears only when the correlations differ in absolute value, and has its own section below.

All three noncentralities condition on the covariate, substituting the expected cross-product matrix for the expectation of its inverse. Inversion is convex, so the assumed sampling variance is too small and every power in the table is overstated, by an amount of order $1/N$. The group test is not exempt: b_{group} is the difference at the covariate mean, and the realized group covariate means are not exactly equal, so the group coefficient inherits their sampling variability even under balanced groups and a covariate independent of them. What the group test's noncentrality does not involve is `sd_cov`, because a correlation is scale free; that is not the same as exactness. The

Type I error rate is unaffected, so it is specifically power that is overstated, not the calibration of the tests. This is the same approximation, and the same convention, that ss_power_c_ancova and ss_power_reg_coef already use, and the correction ss_power_c_ancova names in its own documentation is one instance of it.

The size of that overstatement was measured against simulation for a two-term composite with equal correlations, which isolates the conditioning from everything else, at 200,000 replications per cell: about 2 points of composite power at $n = 25$ per group, under half a point at $n = 50$, and within Monte Carlo error from $n = 100$ up. The constant grows with the number of tests in the composite, because each contributes the same approximation, so a three-term composite is worse than this at any given n. Treat these as the scale of the effect rather than a correction to apply: for planned samples of a few dozen per group the reported power is optimistic by a point or two, and for the sample sizes a three-term composite usually requires the effect has died away.

Because every test divides by the same error estimate, composite power is not monotone in n at the smallest residual degrees of freedom: an error estimate that lands low at one or two degrees of freedom inflates all of the statistics at once, so the composite there can exceed its value at slightly larger n. necessary_n_per_group (or approximate_n_per_group, see the section on unequal residual variances) is the smallest n attaining desired_power, which for a target below alpha_level need not be a size that every larger n also attains.

Value

A data.frame with term and value columns. The design result comes first, then the marginal power and noncentrality of each test in the composite, then rows echoing the planning values, so the assumptions the power was evaluated under travel with the result. The tails row is 2 for nondirectional tests and 1 for directional tests. The names of the composite terms, the implied coefficients, the per-group slopes, and σ_{adj} are carried as attributes rather than rows, keeping the value column numeric. Call plot on the result to draw the population effects.

When the two correlations differ in absolute value the powers are approximations, and the row names say so: composite_power is reported as approximate_composite_power, each power_<term> as approximate_power_<term>, and a planned size as approximate_n_per_group and approximate_N. The section on unequal residual variances explains why. An approximate attribute carries the same flag for a program to test without parsing row names. tidy and glance read both sets of names, so a relabeled table still summarizes to its sample size and its power, and glance() keeps the approximate_ names on the columns it carries through. Nothing changes when the correlations are equal in absolute value, which is the exact case.

Functions

- plot(dmar_composite_power): Draw the population effects a result was planned on, reached from a result already in hand. With show_power = TRUE (the default) the figure is annotated with the power the plan delivers.

Unequal Residual Variances When the Correlations Differ

Within group g the residual variance is $\tau_g^2 = \sigma^2(1 - \rho_g^2)$. Correlations that differ in absolute value therefore leave the two groups with different residual variances, and two things the composite integral assumes stop being true.

The pooled error is no longer one scaled chi square. The residual sum of squares is $\tau_1^2 Q_1 + \tau_2^2 Q_2$ with Q_1 and Q_2 independent chi square variables on $n - 2$ degrees of freedom each, a mixture of two scaled chi squares. Averaging the squared correlations gets its mean exactly right and its spread wrong: the mixture's variance exceeds that of the single scaled chi square the integral uses by $(n-2)(\tau_1^2 - \tau_2^2)^2$, which is zero exactly when the two residual variances agree.

The numerators stop being uncorrelated. The covariate coefficient is the average of the two within-group slopes and the interaction coefficient is their difference, and slopes estimated with different residual variances leave those two estimators correlated: $\mathrm{Cov}(\hat{b}_{cov}, \hat{b}_{group\times cov}) = (\tau_2^2 - \tau_1^2)/(2n\sigma_X^2)$, again zero exactly when the residual variances agree. The tests are then not independent even given the error estimate, which is the step that reduced the composite to a one-dimensional integral in the first place.

What the function reports in that case is therefore the composite power of a design whose pooled error is a single scaled chi square with the right mean and whose tests are conditionally independent, which is a near neighbor of the design described but not that design. It is an approximation, and the output says so: every row carrying a power is renamed with an `approximate_` prefix, and a planned sample size is reported as `approximate_n_per_group` and `approximate_N` rather than as `necessary_n_per_group` and `necessary_N`, because it is the smallest *n* at which the approximation reaches `desired_power`, not an *n* known to attain it. The numbers are the same numbers; only the names change, and only in this case.

The sign of the error is not guaranteed. Two fixed-covariate simulations, run with the covariate values held to the population moments so that the conditioning approximation above plays no part, bracket it. With `smd = 0.30`, `rho = c(0, 0.9)` and $n = 8$ per group, the composite of the group and covariate effects is reported as 0.0752 against a simulated 0.0865 (Monte Carlo standard error 0.0002), so the report is conservative. With `smd = 2.50`, `rho = c(0, 0.95)` and $n = 6$ per group, the group effect alone is reported as 0.9990 against a simulated 0.9979 (Monte Carlo standard error 0.0001), so the report is optimistic. Both are deliberately severe: a correlation gap of 0.9 and a handful of cases per group. At a gap a covariate plausibly shows, `smd = 0.50` with `rho = c(0.1, 0.5)` and $n = 25$ per group, the composite of the group effect and the interaction is reported as 0.1515 against a simulated 0.1508 (Monte Carlo standard error 0.0006), a difference inside simulation error.

What to do with the number, then. Read it as an approximation whose accuracy degrades with the gap between the correlations and not with *N*, and confirm a design you intend to run by simulating it: draw each group's errors with its own residual standard deviation $\sigma\sqrt{1-\rho_g^2}$, fit the same model, and count the replications in which every test in the composite rejects. Equal absolute correlations need none of this. Two correlations of the same magnitude and opposite sign, `rho = c(0.5,-0.5)`, give a large interaction and still equal residual variances, so that design is exact and its rows keep the ordinary names.

Planning Without the Composite

Composite power is the right quantity only when the conclusion needs several results at once. When one effect carries the argument, DMAR already plans for it and this function is unnecessary.

For the group effect, `ss_power_c_ancova` is the planner: a two-group comparison is the contrast `c(1, -1)` on the adjusted means. Naming one term here reproduces it exactly, which is the check the tests assert:

```
ss_power_c_ancova(psi = 0.5, c_weights = c(1, -1), sigma = 1, rho = 0.3,
                  n = 30)
```

```
ss_power_composite_ancova_2group(smd = 0.5, rho = 0.3, n = 30,
                          composite_terms = "group",
                          include_interaction = FALSE)
```

Both return 0.5143. The interaction is dropped in the second call because ss_power_c_ancova plans for the model without it, and carrying a term the other function does not have would spend a residual degree of freedom on nothing. With more than two groups, or a contrast other than a simple difference, ss_power_c_ancova is the only one of the two that applies.

For the design with no covariate at all, ss_power_smd is the planner, and comparing the two is the cleanest way to see what a covariate buys:

```
ss_power_smd(smd = 0.5, n_1 = 30)                    # no covariate
ss_power_c_ancova(psi = 0.5, c_weights = c(1, -1),
                  sigma = 1, rho = 0.5, n = 30)      # covariate, rho = .5
```

Power rises from 0.4779 to 0.5942 because the covariate removes ρ^2 of the error variance, at the cost of one degree of freedom. That is the ANCOVA bargain, and it is worth making before reaching for a composite.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9 on analysis of covariance, and Chapter 3 on statistical power and the noncentral distributions the tests follow.)

See Also

ss_power_composite_ancova for the general one-way or factorial ANCOVA composite (of which this is the two-group case); ss_power_c_ancova for a single contrast on the adjusted means, which is the non-composite planner for this design; ss_power_smd for the two-group design with no covariate; ss_power_c for a contrast with no covariate; ss_power_reg_coef; ci_c_ancova and ci_sc_ancova for intervals on the adjusted means; ancova to fit the model the plan is for

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova
ss_power_composite_factorial_anova(), ss_power_composite_sem(), ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Other composite power: ss_power_composite_ancova(), ss_power_composite_anova(), ss_power_composite_factor
ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_s

Examples

```
# A covariate correlating 0.10 with the outcome in one group and 0.40 in the
# other. The correlations differ, so the slopes differ and there is an
# interaction to detect; nothing had to be assumed about that in advance.
# They also differ in absolute value, so the groups' residual variances
# differ, the powers are approximations, and the rows are named to say so.
ss_power_composite_ancova_2group(smd = 0.20, rho = c(0.10, 0.40), n = 100)

# The covariate is nearly certain to be detected and the interaction is
# better than even, but the group effect is weak, and the composite of all
# three is far below any of them. No single marginal power reveals that.

# Equal correlations mean equal slopes, so the interaction is zero and its
# test rejects only at the Type I error rate. Asking for a composite that
# includes it therefore asks for something that cannot happen often.
ss_power_composite_ancova_2group(smd = 0.20, rho = 0.10, n = 100)

# The composite of the two effects that are actually present.
ss_power_composite_ancova_2group(smd = 0.20, rho = 0.10, n = 100,
                         composite_terms = c("group", "covariate"))

# Per-group sample size for composite power of 0.80 on the group effect and
# the interaction together.
ss_power_composite_ancova_2group(smd = 0.50, rho = c(0.10, 0.50),
                         composite_terms = c("group", "group_by_covariate"),
                         desired_power = 0.80)

# Composite power is not the product of the marginal powers. The tests share
# one error estimate, so they are positively dependent and the composite is
# larger than the product. Multiplying would understate the design. These
# correlations differ in absolute value, so the rows carry the approximate_
# prefix; see the section on unequal residual variances for what that means.
plan <- ss_power_composite_ancova_2group(smd = 0.50, rho = c(0.10, 0.50), n = 95,
                                 composite_terms = c("group",
                                                     "group_by_covariate"))
plan$value[plan$term == "approximate_composite_power"]
prod(plan$value[plan$term %in% c("approximate_power_group",
                                 "approximate_power_group_by_covariate")])

# Correlations of equal magnitude and opposite sign leave the residual
# variances equal, so this design is exact and keeps the ordinary row names,
# even though the two slopes could hardly differ more.
exact <- ss_power_composite_ancova_2group(smd = 0.50, rho = c(0.40, -0.40),
                                  n = 95,
                                  composite_terms = c("group",
                                                      "group_by_covariate"))
exact$value[exact$term == "composite_power"]

# Draw the population effects a result was planned on. The figure needs only
# the effect sizes, and nothing is simulated to draw it.
plot(ss_power_composite_ancova_2group(smd = 0.20, rho = c(0.10, 0.40), n = 100))
```

```
# Planning without the composite: one term returns that test's ordinary
# power, and reproduces ss_power_c_ancova once the interaction is dropped.
ss_power_composite_ancova_2group(smd = 0.50, rho = 0.30, n = 30,
                          composite_terms = "group",
                          include_interaction = FALSE)
ss_power_c_ancova(psi = 0.50, c_weights = c(1, -1), sigma = 1,
                  rho = 0.30, n = 30)

# What the covariate buys, against the same design with no covariate.
ss_power_smd(smd = 0.50, n_1 = 30)

# The broom verbs summarize the plan in one row.
generics::tidy(ss_power_composite_ancova_2group(smd = 0.50, rho = c(0.10, 0.50),
                                        composite_terms = c("group",
                                                            "group_by_covariate"),
                                        desired_power = 0.80))
```

`ss_power_composite_anova`
Sample Size or Composite Power for a One-Way or Factorial ANOVA

Description

Determine the necessary per-cell sample size to achieve a desired level of composite statistical power in a balanced analysis of variance with a groups (a one-way design) or a factorial arrangement of factors, or, given a per-cell sample size, return the realized composite power. Composite power is the probability that every effect named in `effects` is statistically significant in the same study, the quantity a design must be planned against when its conclusion requires more than one result to hold at once. Each effect is a main effect or an interaction, tested by its own F test, and any subset of them can make up the composite.

Usage

```
ss_power_composite_anova(
  factor_levels,
  effects,
  means = NULL,
  sigma = NULL,
  desired_power = 0.85,
  alpha_level = 0.05,
  n_per_cell = NULL
)
```

Arguments

`factor_levels`	Integer vector of the number of levels of each factor, one entry per factor (each at least 2). A single value `a` is a one-way design with a groups; `c(2, 3)` is a 2 by 3 factorial.

effects	A non-empty list naming the effects in the composite. Each element is a list with factors (a vector of factor indices into factor_levels: one index for a main effect, several for an interaction) and, unless means is supplied, exactly one of f (Cohen's *f*) or partial_eta_squared. An optional label names the effect in the output and the figure; the default label is the factor indices joined by x. The purported effect sizes are population values the researcher posits, never sample estimates.
means	Optional array of population cell means whose dimensions are factor_levels (a matrix for two factors), or a numeric vector of length prod(factor_levels) in array order. When supplied, each named effect's Cohen's *f* is computed from the means and sigma, and plot() draws the mean pattern.
sigma	The common within-cell population standard deviation of the outcome, required with means and used only there.
desired_power	Desired composite statistical power (default 0.85). Used only when n_per_cell is NULL.
alpha_level	Type I error rate for each individual *F* test (default 0.05), the per-test rate, not a rate for the composite event.
n_per_cell	Per-cell sample size (balanced); if supplied, the realized composite power is returned rather than a sample size planned.

Details

The population effects can be stated two ways: as effect sizes, a Cohen's *f* or partial eta squared per effect, or as a full array of population cell means together with a common within-cell standard deviation, from which each named effect's *f* is read off the analysis of variance decomposition of the means. Supplying means lets the plot() method draw the mean pattern itself, so the size of the population effects is on the page.

If the design includes one or more covariates, use ss_power_composite_ancova.

Value

A data.frame with term and value columns: the recommended (or supplied) n_per_cell and total N, the composite_power, the residual_df, then for each effect its marginal power_<label>, purported f_<label>, numerator df_<label>, and noncentral_parm_<label>, followed by cells and alpha_level. The result carries the dmar_ss_power class for tidy and glance, and a dmar_composite_power_factori class so plot() draws the figure.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E. (2004). The persistence of underpowered studies in psychological research: Causes, consequences, and remedies. *Psychological Methods, 9*, 147–163.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 7 on factorial designs and Chapter 3 on statistical power.)

See Also

ss_power_composite_ancova for the design with one or more covariates; ss_power_factorial_anova and ss_power_one_way_anova for a single effect

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_factorial_ancova(), ss_power_composite_factoria ss_power_composite_factorial_anova(), ss_power_composite_sem(), ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Other composite power: ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_sem()

Examples

```
# A 2 by 3 factorial ANOVA whose conclusion needs both main effects, so the
# design is planned against their composite.
ss_power_composite_anova(
  factor_levels = c(2, 3),
  effects = list(list(factors = 1, f = 0.25),
                 list(factors = 2, f = 0.20)),
  desired_power = 0.80)

# Realized composite power at 40 per cell for a main effect and the
# interaction of a 2 by 2 design, sizes given as partial eta squared.
ss_power_composite_anova(
  factor_levels = c(2, 2),
  effects = list(list(factors = 1,       partial_eta_squared = 0.06),
                 list(factors = c(1, 2), partial_eta_squared = 0.04)),
  n_per_cell = 40)

# The effects can instead be a full pattern of population cell means with a
# common within-cell SD; plot() then draws the mean pattern itself.
cell_means <- matrix(c(10, 12, 11,
                       13, 12, 16), nrow = 2, byrow = TRUE)
fit <- ss_power_composite_anova(
  factor_levels = c(2, 3), means = cell_means, sigma = 4,
  effects = list(list(factors = 1, label = "A"),
                 list(factors = 2, label = "B")),
  n_per_cell = 30)
fit
plot(fit)
```

ss_power_composite_factorial_ancova
Sample Size or Composite Power for a Factorial ANCOVA

Description

Determine the necessary per-cell sample size to achieve a desired level of composite statistical power in a balanced factorial analysis of covariance, or, given a per-cell sample size, return the realized composite power. Composite power is the probability that every effect named in `effects` is statistically significant in the same study, the quantity a design must be planned against when its conclusion requires more than one result to hold at once. Each effect is a main effect or an interaction of the factorial design, tested by its own F test, and any subset of them can make up the composite.

Usage

```
ss_power_composite_factorial_ancova(
  factor_levels,
  effects,
  means = NULL,
  sigma = NULL,
  covariate_R2 = 0,
  n_covariates = 0,
  desired_power = 0.85,
  alpha_level = 0.05,
  n_per_cell = NULL
)

## S3 method for class 'dmar_composite_power_factorial'
plot(x, ...)
```

Arguments

`factor_levels`	Integer vector of the number of levels of each factor, one entry per factor (each at least 2). A `c(2, 3, 2)` argument is a 2 by 3 by 2 design.
`effects`	A non-empty list naming the effects in the composite. Each element is itself a list with `factors` (a vector of factor indices into `factor_levels`: one index for a main effect, several for an interaction) and exactly one of `f` (Cohen's f for that effect) or `partial_eta_squared`. An optional `label` names the effect in the output and the figure; the default label is the factor indices joined by `x` (for example `"1x2"` for the interaction of factors 1 and 2). Each element also carries the effect size, one of `f` or `partial_eta_squared`, unless `means` is supplied, in which case the sizes come from the means and no effect size is given here. The purported effect sizes are population values the researcher posits, never sample estimates.

means	Optional array of population cell means whose dimensions are factor_levels (a matrix for two factors), or a numeric vector of length prod(factor_levels) in array order (the first factor varying fastest). When supplied, each named effect's Cohen's f is computed from the means and sigma, and plot() draws the mean pattern. The means are the population values the researcher posits, on the raw scale of the outcome.
sigma	The common within-cell population standard deviation of the outcome (the square root of the error variance), required with means and used only there. Cohen's f for an effect is the spread of its cell-mean component relative to sigma.
covariate_R2	Proportion of the outcome's within-cell variance the covariate or covariates explain, in $[0, 1)$. The covariate removes that fraction of the error variance, which raises every effect's noncentrality through $f/\sqrt{1-R^2}$. Defaults to 0.
n_covariates	Number of covariates, a non-negative integer. Each spends one residual degree of freedom. Must be positive when covariate_R2 is. Defaults to 0.
desired_power	Desired composite statistical power (default 0.85). Used only when n_per_cell is NULL.
alpha_level	Type I error rate for each individual F test (default 0.05). This is the per-test rate, not a rate for the composite event.
n_per_cell	Per-cell sample size, assumed balanced across cells; if supplied, the realized composite power is returned rather than a sample size planned.
x	An object returned by ss_power_composite_factorial_ancova or ss_power_composite_factorial_
...	Further arguments to the figure: palette (a palette name, default "okabe_ito") and title.

Details

The population effects can be stated two ways: as effect sizes, a Cohen's f or partial eta squared per effect, or as a full array of population cell means together with a common within-cell standard deviation, from which each named effect's f is read off the analysis of variance decomposition of the means. Supplying means lets the plot() method draw the mean pattern itself, so the size of the population effects is on the page.

With no covariate (the defaults covariate_R2 = 0 and n_covariates = 0) this is a factorial ANOVA; the wrapper ss_power_composite_factorial_anova is that case named directly.

In a balanced factorial design the effect sums of squares are mutually orthogonal, so the effects are uncorrelated. Orthogonal effects do not give independent tests: every F test divides by the same error mean square, so an error estimate that lands low inflates all of the test statistics together. The tests are positively dependent, and composite power is strictly larger than the product of the marginal powers, bounded above by the least powerful test in the set, so the weakest effect governs the design.

Conditional on the error estimate the tests are independent, which reduces the composite to a one dimensional integral over the chi square distribution of that estimate. Effect j has numerator degrees of freedom $\prod(a-1)$ over the factors it spans and noncentrality $N f_{\text{adj}}^2$ with $f_{\text{adj}} = f/\sqrt{1-R^2}$ and N the total sample size, the same convention and covariate adjustment ss_power_factorial_ancova uses. The residual degrees of freedom are $N - \text{cells} - \text{covariates}$. Adaptive quadrature evaluates the

integral, so nothing is simulated and the result is deterministic to quadrature precision. A single-effect composite reproduces the ordinary noncentral F power, and naming one effect reproduces ss_power_factorial_anova (or ss_power_factorial_ancova with a covariate) exactly.

Exactness. With no covariate the composite is exact to quadrature precision: the balanced factorial F tests are exactly noncentral F and exactly independent given the error estimate. A covariate introduces the one approximation ss_power_factorial_ancova already carries, treating covariate_R2 as a fixed reduction of the error variance rather than an estimated one; the departure is of order $1/N$ and is negligible at the sample sizes a multi-effect composite usually needs.

Because every test divides by the same error estimate, composite power is not strictly monotone in n_per_cell at the smallest residual degrees of freedom. necessary_n_per_cell is the smallest per-cell size attaining desired_power.

Value

A data.frame with term and value columns: the recommended (or supplied) n_per_cell and total N, the composite_power, the residual_df, then for each effect its marginal power_<label>, purported f_<label>, numerator df_<label>, and noncentral_parm_<label>, followed by rows echoing covariate_R2, n_covariates, cells, and alpha_level. The result carries the dmar_ss_power class, so tidy and glance summarize the per-cell size and the composite power in broom convention, and a dmar_composite_power_factorial class so plot() draws the figure.

Functions

- plot(dmar_composite_power_factorial): Draw the purported population effect sizes a result was planned on, each annotated with its marginal power, and the composite power in the subtitle.

The figure

The plot() method draws the purported population values. When means were supplied it draws the mean pattern itself: a profile of the population cell means over the first factor, one line per level of the second, faceted by any further factors, with an error bar of plus or minus one within-cell standard deviation at each mean so the effect sizes read against the noise. When effect sizes were supplied instead, the cell means are not pinned (many mean patterns share one Cohen's f), so it draws the effect sizes: one lollipop per named effect at its partial eta squared, colored and labeled by that effect's marginal power. Either way the composite power is in the subtitle and nothing is simulated. Requires **ggplot2**.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E. (2004). The persistence of underpowered studies in psychological research: Causes, consequences, and remedies. *Psychological Methods, 9*, 147–163.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 7 on factorial designs, Chapter 9 on the analysis of covariance, and Chapter 3 on statistical power.)

See Also

ss_power_composite_factorial_anova for the no-covariate case; ss_power_composite_ancova_2group for the two-group ANCOVA composite of the group effect, the covariate effect, and their interaction; ss_power_factorial_ancova and ss_power_factorial_anova for a single effect

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova_he ss_power_composite_factorial_anova(), ss_power_composite_sem(), ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Other composite power: ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_a ss_power_composite_sem()

Examples

```
# A 2 by 3 factorial ANCOVA. The conclusion needs both main effects to hold,
# so the design is planned against their composite. The covariate explains
# 25 percent of the within-cell variance (one covariate).
ss_power_composite_factorial_ancova(
  factor_levels = c(2, 3),
  effects = list(list(factors = 1, f = 0.25),
                 list(factors = 2, f = 0.20)),
  covariate_R2 = 0.25, n_covariates = 1,
  desired_power = 0.80)

# Realized composite power at 40 per cell for a main effect and the
# interaction of a 2 by 2 design, effect sizes given as partial eta squared.
ss_power_composite_factorial_ancova(
  factor_levels = c(2, 2),
  effects = list(list(factors = 1,        partial_eta_squared = 0.06),
                 list(factors = c(1, 2),  partial_eta_squared = 0.04)),
  n_per_cell = 40)

# Naming one effect reproduces the single-effect planner exactly.
ss_power_composite_factorial_ancova(
  factor_levels = c(2, 3), effects = list(list(factors = 2, f = 0.25)),
  n_per_cell = 20)
ss_power_factorial_anova(factor_levels = c(2, 3), effect_indices = 2,
                         f = 0.25, n_per_cell = 20)

# The composite is not the product of the marginal powers. The tests share
# one error estimate, so they are positively dependent and the composite is
# the larger of the two.
plan <- ss_power_composite_factorial_ancova(
  factor_levels = c(2, 2, 3),
  effects = list(list(factors = 1, f = 0.30, label = "A"),
                 list(factors = c(1, 3), f = 0.25, label = "AxC")),
```

```
  n_per_cell = 15)
plan$value[plan$term == "composite_power"]
prod(plan$value[plan$term %in% c("power_A", "power_AxC")])

# A one-row broom summary of the plan.
generics::tidy(ss_power_composite_factorial_ancova(
  factor_levels = c(2, 3),
  effects = list(list(factors = 1, f = 0.25),
                 list(factors = 2, f = 0.20)),
  desired_power = 0.80))

# The figure of the purported population values, annotated with the power a
# given sample size delivers.
plot(ss_power_composite_factorial_ancova(
  factor_levels = c(2, 3),
  effects = list(list(factors = 1, f = 0.25),
                 list(factors = 2, f = 0.20)),
  n_per_cell = 30))

# The effects can instead be stated as a full pattern of population cell means
# with a common within-cell SD. Rows are the 2-level factor, columns the
# 3-level factor. plot() then draws the mean pattern itself.
cell_means <- matrix(c(10, 12, 11,
                       13, 12, 16), nrow = 2, byrow = TRUE)
fit <- ss_power_composite_factorial_ancova(
  factor_levels = c(2, 3), means = cell_means, sigma = 4,
  effects = list(list(factors = 1, label = "A"),
                 list(factors = 2, label = "B")),
  n_per_cell = 30)
fit
plot(fit)
```

`ss_power_composite_factorial_ancova_het` *Composite Power for a Factorial ANCOVA With Heterogeneous Slopes*

Description

Determine the necessary per-cell sample size, or the realized composite power at a supplied per-cell size, for a balanced factorial analysis of covariance in which the covariate's slope may differ across the cells. When the slopes differ, the covariate main effect (the average slope) and the factor-by-covariate slope heterogeneity are themselves testable effects, and any of them, together with the factorial mean effects, can make up the composite. This is the factorial generalization of `ss_power_composite_ancova_2group`, whose two-group model with a correlation per group is the one-factor, two-level case here.

Usage

```
ss_power_composite_factorial_ancova_het(
```

```
  factor_levels,
  effects,
  means = NULL,
  correlations = NULL,
  sigma = NULL,
  sd_cov = 1,
  desired_power = 0.85,
  alpha_level = 0.05,
  n_per_cell = NULL
)

## S3 method for class 'dmar_composite_power_factorial_het'
plot(x, ...)
```

Arguments

factor_levels	Integer vector of the number of levels of each factor (each at least 2).
effects	A non-empty list naming the effects in the composite. Each element has a type, one of "mean" (a factorial main effect or interaction on the cell means; the default), "covariate" (the average covariate slope), or "slope" (a factor-by-covariate slope heterogeneity). A "mean" or "slope" effect also gives factors, the factor indices it spans; a "covariate" effect spans none. Each element carries its effect size, f or partial_eta_squared, unless the population values (means and correlations) are supplied, in which case the sizes come from them. An optional label names the effect; defaults are the factor indices joined by x for a mean effect, "covariate", and "cov_x_<factors>" for a slope effect.
means	Optional array of population cell means (dimensions factor_levels), needed when a "mean" effect is in the composite. Supplies the mean effects' sizes.
correlations	Optional array of the population covariate-outcome correlation within each cell (dimensions factor_levels, each in $(-1, 1)$). The cell slopes it implies supply the covariate and slope effects' sizes, and its spread across cells sets the slope heterogeneity. Required with the population-values interface, since the pooled error depends on every cell's correlation.
sigma	The common within-cell population standard deviation of the outcome (before adjustment), required with the population-values interface.
sd_cov	Population standard deviation of the covariate. Default 1. A correlation is scale free, so sd_cov does not change any power; it sets the units the slopes and the figure are drawn in.
desired_power	Desired composite statistical power (default 0.85). Used only when n_per_cell is NULL.
alpha_level	Type I error rate for each individual test (default 0.05).
n_per_cell	Per-cell sample size (balanced); if supplied, the realized composite power is returned rather than a sample size planned.
x	An object returned by ss_power_composite_factorial_ancova_het.
...	Further arguments to the figure: palette, title, and, for the regression-line figure, cov_range (covariate range in SDs either side of the mean, default 2).

Details

Kept separate from ss_power_composite_factorial_ancova, which assumes one common slope and treats the covariate only as a variance reducer. Use this function when the covariate slope is expected to differ across conditions, or when the test of that difference is part of the design.

The model fits the factorial mean structure, the covariate, and every factor-by-covariate slope term, so it has $2 \times \text{cells}$ parameters and residual degrees of freedom $N - 2\,\text{cells}$. Under balance and a covariate with a common distribution across the cells, those terms are mutually orthogonal, so the tests share only the pooled residual and the composite is the shared-error integral of ss_power_composite_factorial_ancova. A "mean" effect on factor set S has numerator df $\prod(a-1)$ and is read from the cell means against the pooled adjusted error $\sigma^2(1-\bar{\rho}^2)$; a "covariate" effect is the grand slope on 1 df; a "slope" effect on S is the slope heterogeneity across those factors, with the same df as the matching mean effect, read from the cell slopes.

The two approximations are those of ss_power_composite_ancova_2group. The covariate-related tests condition on the covariate cross-products, overstating power by an amount of order $1/N$, which is negligible past a few dozen per cell. Separately, correlations that differ in absolute value across cells make the pooled error a mixture of scaled chi squares rather than the single one the integral assumes; that one does not shrink with *N* and is the subject of the section below.

Value

A data.frame with term and value columns: the recommended (or supplied) n_per_cell and total N, the composite_power, the residual_df, then for each effect its marginal power_<label>, purported f_<label>, numerator df_<label>, and noncentral_parm_<label>, followed by cells and alpha_level. Carries the dmar_ss_power class for tidy / glance and a dmar_composite_power_factorial_het class for plot().

When the supplied cell correlations differ in absolute value the powers are approximations and the row names say so, with an approximate_ prefix on every power and on a planned sample size; see the section on unequal residual variances. An approximate attribute carries the same flag for a program to test without parsing row names, and tidy and glance read both sets of names, so a relabeled table still summarizes to its per-cell size and its composite power.

Functions

- plot(dmar_composite_power_factorial_het): Draw the purported population values: the per-cell regression lines when population values were supplied, or the effect size lollipop otherwise.

Unequal Residual Variances When the Cell Correlations Differ

In cell c the residual variance is $\tau_c^2 = \sigma^2(1-\rho_c^2)$, so cell correlations that differ in absolute value leave the cells with different residual variances. Averaging the squared correlations, which is what $\sigma^2(1-\bar{\rho}^2)$ does, gets the expected pooled error right and its distribution wrong: the residual sum of squares is $\sum_c \tau_c^2 Q_c$ with the Q_c independent chi square variables, a mixture of scaled chi squares with the same mean and a larger spread than the single scaled chi square the shared-error integral integrates over. The numerators are affected too. The covariate effect is the average cell slope and a slope effect is a contrast among the cell slopes, and cell slopes estimated with different residual variances give estimators of those two that are correlated, so the tests are not independent even given the error estimate.

Every power the function reports is then an approximation, and the output says so. `composite_power` is reported as `approximate_composite_power`, each `power_<label>` as `approximate_power_<label>`, and a planned sample size as `approximate_n_per_cell` and `approximate_N`, because that size is the smallest one at which the approximation reaches `desired_power` rather than a size known to attain it. The numbers do not change; the names do, and only in this case. Equal absolute cell correlations, including cells whose correlations share a magnitude and differ in sign, are the exact case and keep the ordinary names.

The error has no guaranteed sign, and it grows with the spread of the cell correlations rather than shrinking with *N*. Treat the number as an approximation and confirm a design you intend to run by simulating it: draw each cell's errors with its own residual standard deviation $\sigma\sqrt{1-\rho_c^2}$, fit the same model, and count the replications in which every effect in the composite is significant. The two-group help page reports the size of the departure over a range of correlation gaps.

The effect size interface is a special case worth naming. Supplying `f` or `partial_eta_squared` for each effect states the effects directly and says nothing about the cell correlations, so the function has nothing to detect unequal residual variances from and labels the table exact. If the design those effect sizes came from has cell correlations that differ in absolute value, the same approximation applies and the labels will not tell you; supply `correlations` instead when you want the function to keep track of it.

The figure

When the population values are supplied, `plot()` draws the population regression line in each cell over the covariate, so heterogeneous slopes show as lines of different angle and mean effects as vertical separation, colored by the first factor and faceted by any others. When effect sizes are supplied instead, it draws the effect size lollipop of `ss_power_composite_factorial_ancova`. Either way the composite power is in the subtitle. Requires **ggplot2**.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9 on the analysis of covariance and heterogeneity of regression, and Chapter 7 on factorial designs.)

See Also

`ss_power_composite_factorial_ancova` for the common-slope version; `ss_power_composite_ancova_2group` for the two-group case

Other sample size for power: `power_fisher_exact()`, `ss_aipe_mixed_effects()`, `ss_power_R2()`, `ss_power_R2_sensitivity()`, `ss_power_c()`, `ss_power_c_ancova()`, `ss_power_composite_ancova()`, `ss_power_composite_ancova_2group()`, `ss_power_composite_anova()`, `ss_power_composite_factorial_ancova()`, `ss_power_composite_factorial_anova()`, `ss_power_composite_sem()`, `ss_power_contrast()`, `ss_power_equivalence_c()`, `ss_power_factorial_ancova()`, `ss_power_factorial_anova()`, `ss_power_indirect_effect()`, `ss_power_mixed_effects()`, `ss_power_one_way_anova()`, `ss_power_pcm()`, `ss_power_r()`, `ss_power_rc()`, `ss_power_reg_coef()`, `ss_power_reg_coef_sensitivity()`, `ss_power_rm_anova()`, `ss_power_sc()`, `ss_power_sem()`, `ss_power_smd()`, `ss_power_split_plot_anova()`

Other composite power: ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_anova(), ss_power_composite_sem()

Examples

```
# A 2 by 2 design whose conclusion needs a main effect and evidence that the
# covariate slope differs across the first factor. Effect sizes stated
# directly: the main effect (mean), the average covariate effect, and the
# factor 1 by covariate slope heterogeneity.
ss_power_composite_factorial_ancova_het(
  factor_levels = c(2, 2),
  effects = list(list(type = "mean",      factors = 1,  f = 0.25),
                 list(type = "covariate",               f = 0.40),
                 list(type = "slope",      factors = 1,  f = 0.20)),
  desired_power = 0.80)

# The same design from population values: cell means, a covariate-outcome
# correlation per cell (they differ across factor 1, so the slopes do), and a
# common within-cell SD. plot() then draws the per-cell regression lines.
cell_means <- matrix(c(10, 12,
                       11, 13), nrow = 2, byrow = TRUE)
cell_rho   <- matrix(c(0.55, 0.55,
                       0.15, 0.20), nrow = 2, byrow = TRUE)
fit <- ss_power_composite_factorial_ancova_het(
  factor_levels = c(2, 2), means = cell_means, correlations = cell_rho,
  sigma = 4, sd_cov = 2,
  effects = list(list(type = "mean",      factors = 1),
                 list(type = "covariate"),
                 list(type = "slope",      factors = 1)),
  n_per_cell = 40)
fit
plot(fit)

# The one-factor, two-level case is the two-group composite ANCOVA.
ss_power_composite_factorial_ancova_het(
  factor_levels = 2,
  means = c(-0.25, 0.25), correlations = c(0.1, 0.5), sigma = 1,
  effects = list(list(type = "mean", factors = 1),
                 list(type = "slope", factors = 1)),
  n_per_cell = 100)
```

---

ss_power_composite_factorial_anova
Sample Size or Composite Power for a Factorial ANOVA

Description

Determine the necessary per-cell sample size to achieve a desired level of composite statistical power in a balanced factorial analysis of variance, or, given a per-cell sample size, return the realized composite power. Composite power is the probability that every effect named in `effects` is statistically significant in the same study, the quantity a design must be planned against when its conclusion requires more than one result to hold at once.

Usage

```
ss_power_composite_factorial_anova(
  factor_levels,
  effects,
  means = NULL,
  sigma = NULL,
  desired_power = 0.85,
  alpha_level = 0.05,
  n_per_cell = NULL
)
```

Arguments

`factor_levels`	Integer vector of the number of levels of each factor, one entry per factor (each at least 2). A `c(2, 3, 2)` argument is a 2 by 3 by 2 design.
`effects`	A non-empty list naming the effects in the composite; see `ss_power_composite_factorial_ancova` for the format. Each element gives the `factors` an effect spans and its `f` or `partial_eta_squared`, unless `means` is supplied.
`means`	Optional array of population cell means (dimensions `factor_levels`), or a numeric vector of length `prod(factor_levels)` in array order, from which each effect's Cohen's f is read given `sigma`. When supplied, `plot()` draws the mean pattern. See `ss_power_composite_factorial_ancova`.
`sigma`	The common within-cell standard deviation of the outcome, required with `means` and used only there.
`desired_power`	Desired composite statistical power (default 0.85). Used only when `n_per_cell` is `NULL`.
`alpha_level`	Type I error rate for each individual F test (default 0.05), the per-test rate rather than a rate for the composite event.
`n_per_cell`	Per-cell sample size, assumed balanced; if supplied, the realized composite power is returned rather than a sample size planned.

Details

This is the no-covariate case of `ss_power_composite_factorial_ancova`, named directly. It does not take a covariate; when a covariate belongs in the model, use `ss_power_composite_factorial_ancova`, which raises every effect's power for the variance the covariate explains. With no covariate the composite is exact to quadrature precision, since the balanced factorial F tests are exactly noncentral F and exactly independent given the shared error estimate.

See ss_power_composite_factorial_ancova for the model, the one-dimensional integral that evaluates the composite over the shared error estimate, and the figure the plot() method draws. Naming one effect reproduces ss_power_factorial_anova exactly.

Value

A data.frame with term and value columns, as ss_power_composite_factorial_ancova returns but with covariate_R2 0 and n_covariates 0. It carries the dmar_ss_power class for tidy / glance and the dmar_composite_power_factorial class for plot().

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E. (2004). The persistence of underpowered studies in psychological research: Causes, consequences, and remedies. *Psychological Methods, 9*, 147–163.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 7 on factorial designs and Chapter 3 on statistical power.)

See Also

ss_power_composite_factorial_ancova for the version that admits a covariate; ss_power_factorial_anova for a single effect

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_sem(), ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Other composite power: ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova
ss_power_composite_sem()

Examples

```
# A 2 by 3 factorial ANOVA whose conclusion needs both main effects. Plan
# against their composite, not against either one alone.
ss_power_composite_factorial_anova(
  factor_levels = c(2, 3),
  effects = list(list(factors = 1, f = 0.25),
                 list(factors = 2, f = 0.20)),
  desired_power = 0.80)

# Realized composite power at 25 per cell for a main effect and the
```

```
# three-way interaction of a 2 by 2 by 2 design.
ss_power_composite_factorial_anova(
  factor_levels = c(2, 2, 2),
  effects = list(list(factors = 1,          f = 0.30, label = "A"),
                 list(factors = c(1, 2, 3), f = 0.25, label = "AxBxC")),
  n_per_cell = 25)

# Naming one effect reproduces the single-effect planner exactly.
ss_power_composite_factorial_anova(
  factor_levels = c(2, 3), effects = list(list(factors = 2, f = 0.25)),
  n_per_cell = 20)
ss_power_factorial_anova(factor_levels = c(2, 3), effect_indices = 2,
                         f = 0.25, n_per_cell = 20)

# The figure of the purported population effect sizes.
plot(ss_power_composite_factorial_anova(
  factor_levels = c(2, 3),
  effects = list(list(factors = 1, f = 0.25),
                 list(factors = 2, f = 0.20)),
  n_per_cell = 30))

# Stating the effects as population cell means with a common within-cell SD.
# plot() then draws the mean pattern, with error bars of one SD.
cell_means <- matrix(c(10, 12, 11,
                       13, 12, 16), nrow = 2, byrow = TRUE)
plot(ss_power_composite_factorial_anova(
  factor_levels = c(2, 3), means = cell_means, sigma = 4,
  effects = list(list(factors = 1, label = "A"),
                 list(factors = 2, label = "B")),
  n_per_cell = 30))
```

`ss_power_composite_sem`

Sample Size or Composite Power for a Set of SEM Parameters

Description

Determine the necessary sample size for a structural equation model study so that every parameter of interest is statistically significant in the same study with a desired probability, or, given a sample size, return that probability. Composite power is the probability that all of the named parameters are significant jointly, the quantity a design must be planned against when its conclusion requires more than one result to hold at once: a study can have adequate power for each hypothesis on its own and still be underpowered for the conclusion that rests on all of them together (Maxwell, 2004). The parameters of interest are any labeled parameters of a lavaan analysis model, structural paths, loadings, covariances, or quantities defined with `:=` such as an indirect effect, and any subset of them can make up the composite.

Usage

```
ss_power_composite_sem(
  model,
  Sigma = NULL,
  pop_model = NULL,
  mu = NULL,
  parameters = NULL,
  desired_power = 0.85,
  alpha_level = 0.05,
  N = NULL,
  G = 1000,
  seed = NULL,
  ...
)
```

Arguments

model	A single character string giving the free analysis model in lavaan model syntax (see model.syntax), the model that would be fit to the data. Each parameter of interest must carry a parameter label so it can be referred to by name, for example "f2 ~ b*f1" labels the structural path b, and "ab := a*b" defines an indirect effect from the labeled paths a and b.
Sigma	Population covariance matrix of the observed variables, with row and column names matching the observed variables in model. It is typically obtained from a fully fixed population model via cov_sem. Supply exactly one of Sigma or pop_model.
pop_model	A single character string giving the population model in lavaan model syntax with every parameter fixed to its population value, from which cov_sem derives Sigma (and the population means, when the model has a mean structure). This is where the purported population values of the parameters of interest are chosen; they are values the researcher posits (from theory, prior studies, or pilot data), never sample estimates. Supply exactly one of Sigma or pop_model.
mu	Optional population means of the observed variables, used with Sigma: a named numeric vector with one entry per observed variable, or an unnamed vector in the row order of Sigma. The default NULL is zero means. Means matter only when the analysis model has a mean structure (an intercept term such as s ~ 1, as in a latent growth curve model); when pop_model is supplied its mean structure provides the means and mu must not also be given.
parameters	Character vector of the parameter labels that make up the composite. The default NULL uses every user-labeled parameter in model, in order of appearance, so labeling exactly the parameters of interest is the simplest way to state the set.
desired_power	Desired composite statistical power (default 0.85). Used only when N is NULL.
alpha_level	Type I error rate for each individual two-sided Wald z test (default 0.05), the per-test rate, not a rate for the composite event.
N	Sample size; if supplied, the realized composite power at that N is returned rather than a sample size planned.

G — Number of converged Monte Carlo replications per evaluated sample size (default 1000). The simulation error of each estimated power is about $\sqrt{p(1-p)/G}$; raise G for a sharper answer.

seed — Optional integer seed for reproducibility. The default NULL uses the current state of the random number generator; a supplied seed is set internally and the prior state restored on exit.

... — Additional arguments passed to sem, both when the population values are resolved and for every Monte Carlo fit (for example std.lv = TRUE or missing = "listwise"). A robust estimator such as "MLM" cannot be used here: the same arguments reach the setup fit, which is always from summary statistics, and lavaan refuses a robust estimator there. The Monte Carlo data are multivariate normal by construction, so a robust estimator would buy nothing.

Details

Analytic sample size planning methods in SEM exist for a single targeted parameter (Satorra & Saris, 1985; Lai & Kelley, 2011) or for overall model fit (MacCallum, Browne, & Sugawara, 1996; ss_power_sem), but most studies state several hypotheses and support their conclusion only when all of them hold. This function plans for that case by a priori Monte Carlo simulation (Muthén & Muthén, 2002; Maxwell, Kelley, & Rausch, 2008): for a candidate *N*, G data sets are drawn from the multivariate normal population with covariance matrix Sigma, the analysis model is fit to each, and each parameter of interest is tested with its two-sided Wald z test at alpha_level. The proportion of replications in which every parameter is significant estimates the composite power, and the per-parameter proportions estimate the marginal powers. Because the estimates share one fitted model, the tests are dependent; the simulation reflects that dependence exactly, at the stated *N*, with no asymptotic shortcut.

The composite event is contained in each marginal event, so composite power is at most the smallest marginal power: the weakest parameter governs the design, and the marginal power_<label> rows show which parameter that is.

When N is NULL the necessary sample size is searched for. The search starts where the product of the marginal Wald powers (an independence approximation computed from the asymptotic variances, spent before any simulation) reaches desired_power, brackets the crossing geometrically, and bisects to adjacent integers, each candidate evaluated with its own G replications. A planning call therefore fits the analysis model several thousand times at the default G, and even at the smallest admissible G the search runs for several seconds, so the example below evaluates a stated N, which is the cheap half of the method. A planning call is the same call with N left out and desired_power stated or left at its default, and the first row of the result is then necessary_N rather than specified_N. The vignette vignette("composite_sem_planning", package = "DMAR") works through the planning calls for a mediation model and a latent growth curve model, with reference values computed at G = 10000.

Value

A data.frame with term and value columns: the necessary_N (or supplied specified_N), the composite_power and its simulation standard error composite_power_mc_se, then for each parameter its marginal power_<label> and purported population_<label> value under the analysis model, followed by alpha_level, the requested replications, the converged_replications the summary is based on, and, when a size was planned, the desired_power. The result carries the

dmar_ss_power class, so tidy and glance summarize the sample size and the composite power in broom convention.

Monte Carlo Precision

Each reported power is a proportion of G replications, with simulation standard error about $\sqrt{p(1-p)/G}$; the composite_power_mc_se row reports it for the composite. The necessary sample size inherits that uncertainty: near the target the power curve is flat enough that neighboring N are separated by less than the simulation error, so repeated calls with different seeds return slightly different sizes. Raising G narrows the spread; reporting the seed makes a plan reproducible. A proportion of G replications takes only the values $0, 1/G, \ldots, 1$, so a desired_power above $1 - 1/G$ is refused with a message saying how large G must be for that target; the same resolution guard applies to ss_aipe_composite_sem's assurance.

Note

A replication whose fit does not converge, or converges without a usable standard error for some parameter of interest, is discarded and fresh data are drawn, up to 20 * G attempts per evaluated sample size; the reported powers condition on convergence. When fewer than G replications converge within the cap, a single warning is issued and the summary is based on the converged replications (their count is the converged_replications row). Frequent nonconvergence at small N is itself design information: a sample size at which the model rarely converges is too small in a sense that precedes power.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Lai, K., & Kelley, K. (2011). Accuracy in parameter estimation for targeted effects in structural equation modeling: Sample size planning for narrow confidence intervals. *Psychological Methods, 16*(2), 127–148. doi:10.1037/a0021764

MacCallum, R. C., Browne, M. W., & Sugawara, H. M. (1996). Power analysis and determination of sample size for covariance structure modeling. *Psychological Methods, 1*(2), 130–149. doi:10.1037/1082989X.1.2.130

Maxwell, S. E. (2004). The persistence of underpowered studies in psychological research: Causes, consequences, and remedies. *Psychological Methods, 9*(2), 147–163. doi:10.1037/1082989X.9.2.147

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on statistical power.)

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

Muthén, L. K., & Muthén, B. O. (2002). How to use a Monte Carlo study to decide on sample size and determine power. *Structural Equation Modeling, 9*(4), 599–620. doi:10.1207/S15328007SEM0904_8

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48*(2), 1–36. doi:10.18637/jss.v048.i02

Satorra, A., & Saris, W. E. (1985). Power of the likelihood ratio test in covariance structure analysis. *Psychometrika, 50*(1), 83–90.

See Also

ss_aipe_composite_sem for the same set of parameters planned for accuracy in parameter estimation (AIPE) instead of significance; cov_sem for deriving Sigma from a fully fixed population model; ss_power_sem for overall model fit; ss_aipe_sem_path for a single targeted path; ss_power_composite_anova and its siblings for composite power in ANOVA and ANCOVA designs, where the composite is evaluated by quadrature rather than simulation.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Other composite power: ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova, ss_power_composite_factorial_anova()

Examples

```
# A three-factor model whose conclusion rests on three structural paths
# at once: f1 predicting f2, f2 predicting f3, and f1 predicting f3
# directly. The population model fixes every parameter to its purported
# population value.
pop_model <- "
  f1 =~ 1*y1 + 0.8*y2 + 0.8*y3
  f2 =~ 1*y4 + 0.8*y5 + 0.8*y6
  f3 =~ 1*y7 + 0.8*y8 + 0.8*y9
  f2 ~ 0.4*f1
  f3 ~ 0.3*f2 + 0.25*f1
  f1 ~~ 1*f1
  f2 ~~ 0.84*f2
  f3 ~~ 0.8*f3
  y1 ~~ 0.5*y1; y2 ~~ 0.5*y2; y3 ~~ 0.5*y3
  y4 ~~ 0.5*y4; y5 ~~ 0.5*y5; y6 ~~ 0.5*y6
  y7 ~~ 0.5*y7; y8 ~~ 0.5*y8; y9 ~~ 0.5*y9
"

# The analysis model is free; the labels name the parameters of interest.
analysis_model <- "
  f1 =~ y1 + y2 + y3
  f2 =~ y4 + y5 + y6
  f3 =~ y7 + y8 + y9
  f2 ~ a*f1
  f3 ~ b*f2 + c*f1
"
```

```
# Realized composite power at N = 200. The probability that all three
# paths come out significant in the same study is lower than the marginal
# power of any one of them: the composite event sits inside each marginal
# event, so the weakest parameter governs the design. G = 20 keeps the
# example quick; a reported plan deserves the default G = 1000 or more.
ss_power_composite_sem(model = analysis_model, pop_model = pop_model,
                       N = 200, G = 20, seed = 113)
```

`ss_power_contrast` *Sample Size and Statistical Power for a Contrast in a Fixed-Effects ANOVA*

Description

Determine the necessary per-group sample size for a contrast in a one-way fixed-effects ANOVA so as to achieve a desired level of statistical power, or, alternatively, compute the achieved power for a given sample size. The contrast is specified by a vector of weights and the population means (or a population contrast value) along with the within-group variance.

Usage

```
ss_power_contrast(
  c_weights,
  mu = NULL,
  sigma_squared = NULL,
  psi = NULL,
  desired_power = 0.85,
  alpha_level = 0.05,
  directional = FALSE,
  n_per_group = NULL,
  print_progress = FALSE
)
```

Arguments

`c_weights`	Vector of contrast weights. Required to satisfy `sum(c_weights) == 0`, `sum(c_weights[c_weights > 0]) == 1`, and `sum(c_weights[c_weights < 0]) == -1`; that is, write the contrast in normalized fractional form so that the positive coefficients sum to 1 and the negative coefficients sum to -1. The contrast estimate is then directly interpretable as a (weighted) mean difference.
`mu`	Vector of population group means, of length `length(c_weights)`. Either `mu` or `psi` must be supplied (but not both).
`sigma_squared`	Population within-group variance (σ^2); must be a positive number.
`psi`	Optional. Directly specify the population contrast value $\psi = \sum_j c_j \mu_j$ as a single number. Use this when the means themselves are not of interest, only the contrast value.

`desired_power`	Target statistical power for the test of the contrast (default `0.85`). Used only when `n_per_group` is `NULL`.
`alpha_level`	Type I error rate (default `0.05`).
`directional`	Logical. `FALSE` (default) yields a two-sided test. `TRUE` yields a one-sided test in the direction of the population contrast.
`n_per_group`	Optional per-group sample size at which to evaluate power. May be a scalar (the common per-group size, equal across groups) or a numeric vector of length `length(c_weights)` giving each group's size. When `NULL` (default), the function instead solves for the smallest per-group n achieving `desired_power`.
`print_progress`	If `TRUE`, print the trial n and the corresponding power as the iterative search proceeds.

Details

Let $\psi = \sum_j c_j \mu_j$ be the population contrast. Under the usual fixed-effects ANOVA model with common within-group variance σ^2 and per-group sizes n_j, the standard error of the contrast estimate is

$$\mathrm{SE}_{\hat{\psi}} = \sqrt{\sigma^2 \sum_j c_j^2 / n_j}\,,$$

the test statistic $t = \hat{\psi}/\mathrm{SE}_{\hat{\psi}}$ follows a central t distribution with $N - a$ degrees of freedom under $H_0 : \psi = 0$, and a noncentral t distribution with the same df and noncentrality parameter $\lambda = \psi/\mathrm{SE}_{\hat{\psi}}$ under the alternative.

For a two-sided test at level α,

$$\text{Power} = \Pr(t > t_{1-\alpha/2,\,df}) + \Pr(t < -t_{1-\alpha/2,\,df}),$$

computed exactly from the noncentral t distribution; for a one-sided test, only the appropriate tail contributes.

Cohen's f for a one-df contrast is reported as $f = |\lambda|/\sqrt{N}$, equivalently $f^2 = F_{\text{pop}}/N$ (Cohen, 1988, Ch. 8); this matches the value that `pwr.f2.test` expects when used with `u = 1` and `v = N - a`.

Value

A `data.frame` with two columns, `term` and `value`:

When solving for *n* (`n_per_group = NULL`) Rows are `necessary_n_per_group`, `total_N`, `actual_power`, `noncentral_t_parm`, and `effect_size_f` (Cohen's f for a one-degree-of-freedom contrast).

When evaluating power (`n_per_group` **supplied**) Rows are `specified_n_per_group` (or `NA` for unequal n), `total_N`, `actual_power`, `noncentral_t_parm`, and `effect_size_f`.

The result carries the `dmar_ss_power` class, so `tidy` and `glance` summarize it in broom convention; the reported size is the per-group n (or `NA` when unequal group sizes are supplied).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

contrast_test, ss_power_reg_coef, cv_t

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_sc ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# Four-group example from Maxwell & Delaney's textbook tradition: contrast
# the average of three treatment means with a fourth (control), under
# population means (90, 92, 88, 81), within-group variance 144.
#
# 1. Power achieved at n = 20 per group (total N = 80). Should be ~ .80.
ss_power_contrast(
  c_weights     = c(1/3, 1/3, 1/3, -1),
  mu            = c(90, 92, 88, 81),
  sigma_squared = 144,
  n_per_group   = 20
)

# 2. Per-group sample size needed for power = .90.
ss_power_contrast(
  c_weights     = c(1/3, 1/3, 1/3, -1),
  mu            = c(90, 92, 88, 81),
  sigma_squared = 144,
  desired_power = 0.90
)

# 3. Same effect size specification using a directly given psi.
ss_power_contrast(
  c_weights     = c(1/3, 1/3, 1/3, -1),
  psi           = 9,
```

```
  sigma_squared = 144,
  n_per_group   = 20
)

# 4. Unequal per-group sample sizes.
ss_power_contrast(
  c_weights     = c(0.5, 0.5, -0.5, -0.5),
  mu            = c(90, 92, 88, 81),
  sigma_squared = 144,
  n_per_group   = c(15, 25, 25, 15)
)
```

`ss_power_equivalence_c` *Sample Size for Equivalence or Noninferiority of a Linear Contrast*

Description

Computes the smallest per-group sample size at which the two one-sided tests procedure (Schuirmann, 1987) for a linear contrast $\psi = \sum_j c_j \mu_j$, or the companion one-sided noninferiority test, attains a desired power. Power is computed exactly through `power_equivalence_c`. This is the declaration-probability route to planning; the accuracy in parameter estimation (AIPE) route, which targets the confidence interval width directly, is `ss_aipe_c`, and the two answer the same question whenever `true_psi = 0` and the width target is calibrated to the bounds.

Usage

```
ss_power_equivalence_c(
  c_weights,
  sigma,
  delta_lower = NULL,
  delta_upper = NULL,
  true_psi = 0,
  desired_power = 0.85,
  alpha_level = 0.05,
  side = c("equivalence", "noninferiority")
)
```

Arguments

`c_weights`	The contrast weights. The weights must sum to zero with the positive weights summing to 1 and the negative weights to -1, so that the bounds are on the raw scale of the response.
`sigma`	The anticipated error standard deviation (the square root of the mean square error).

`delta_lower`, `delta_upper`
: Equivalence bounds on the raw scale of the response. Both must be positive; the equivalence region is $(-\delta_L, +\delta_U)$. If only `delta_upper` is supplied, the bounds are symmetric. Noninferiority uses $-\delta_L$ alone.

`true_psi`
: The population contrast the design should be able to detect as equivalent (or noninferior). Default `0`. For equivalence it must lie strictly inside the bounds; for noninferiority, strictly above $-\delta_L$. The farther `true_psi` sits from the center of the region, the larger the required sample size.

`desired_power`
: The target probability of declaring equivalence (or noninferiority) at `true_psi`. Default `0.85`, matching `ss_power_contrast`.

`alpha_level`
: One-sided significance level for each test. Default `0.05`.

`side`
: `"equivalence"` (default) or `"noninferiority"`.

Details

Design. Planning assumes equal allocation across the J groups named by `c_weights` and a pooled error term on $N - J$ degrees of freedom. Groups with zero weight still contribute error degrees of freedom, which is why they belong in `c_weights` when the fitted model will include them.

The search. The function starts from the normal-theory approximation and moves to the smallest integer n whose exact power reaches `desired_power`. Power is monotone in n once the design is feasible, so the search is a short walk.

Relation to the half-width rule. With symmetric bounds $\pm\delta$, `true_psi = 0`, and $\alpha = .05$, targeting a 90% CI half-width of $\delta/2$ yields a declaration probability of about .90; this function makes the probability the target directly rather than through the width.

Value

A `data.frame` with rows `necessary_n_per_group` (the recommended sample size for each of the J groups named by `c_weights`), `total_N` (the implied total, $J \times n$), and `actual_power` (the exact power achieved at the recommendation). The result carries the `dmar_ss_power` class, so `tidy` and `glance` summarize it in broom convention.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Chattopadhyay, B., Bandyopadhyay, T., Kelley, K., & Padalunkal, J. J. (2025). A sequential approach for noninferiority or equivalence of a linear contrast under cost constraints. *Psychological Methods, 30*(2), 425–439. doi:10.1037/met0000570

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics, 15*(6), 657–680.

See Also

power_equivalence_c, ss_aipe_c, ss_power_contrast, equivalence_c

Other equivalence testing: equivalence_c(), equivalence_r(), equivalence_smd(), plot_equivalence(), power_density_equivalence_md(), power_equivalence_c(), power_equivalence_md(), power_equivalence_md_plo

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_sc ss_power_contrast(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# 1. Two-group equivalence with bounds of 5 raw-scale points and an
#    anticipated error SD of 15.67: n per group for 90% power at
#    true equivalence.
ss_power_equivalence_c(c_weights = c(1, -1), sigma = 15.67,
                       delta_upper = 5, desired_power = 0.90)

# 2. Noninferiority is cheaper than equivalence at the same bound.
ss_power_equivalence_c(c_weights = c(1, -1), sigma = 15.67,
                       delta_upper = 5, desired_power = 0.90,
                       side = "noninferiority")

# 3. A true contrast off center raises the requirement.
ss_power_equivalence_c(c_weights = c(1, -1), sigma = 15.67,
                       delta_upper = 5, true_psi = 2,
                       desired_power = 0.90)
```

`ss_power_factorial_ancova`

Sample Size Planning for Power in Factorial ANCOVA

Description

Power and sample size for any effect (a main effect or any interaction) in a between-subjects factorial design with covariates: the analysis of covariance generalization of ss_power_factorial_anova. Covariates earn their keep by absorbing error variance: with a joint squared multiple correlation R^2 between the covariates and the outcome within cells, the error variance falls by the factor $1 - R^2$, so an effect size f defined on the original (ANOVA) metric grows to $f/\sqrt{1 - R^2}$ in the covariate-adjusted analysis, at the price of one error degree of freedom per covariate.

Usage

```
ss_power_factorial_ancova(
  factor_levels,
  effect_indices,
  f = NULL,
  partial_eta_squared = NULL,
  covariate_R2 = 0,
  n_covariates = 0,
  desired_power = 0.85,
  alpha_level = 0.05,
  n_per_cell = NULL
)
```

Arguments

`factor_levels`	Integer vector giving the number of levels of each factor, for example `c(2, 4, 3)` for a 2 x 4 x 3 design.
`effect_indices`	Integer vector identifying the factors that define the effect of interest: `1` for the first factor's main effect, `c(2, 3)` for the B x C interaction, and so on.
`f`	Cohen's f for the chosen effect *on the unadjusted (ANOVA) metric*, that is, with the within-cell standard deviation of the outcome in its denominator. Supply this or `partial_eta_squared`, not both.
`partial_eta_squared`	Partial eta squared for the chosen effect on the unadjusted metric.
`covariate_R2`	Joint squared multiple correlation between the covariates and the outcome within cells, in $[0, 1)$. `0` reproduces the ANOVA analysis (with the covariate degrees of freedom still spent if `n_covariates > 0`).
`n_covariates`	Number of covariates, a non-negative integer.
`desired_power`	Desired power; the per-cell sample size is solved when `n_per_cell` is `NULL`. Defaults to 0.85 to match `ss_power_factorial_anova`.
`alpha_level`	Type I error rate.
`n_per_cell`	Per-cell sample size; when supplied, the realized power at that size is returned instead of solving for size.

Details

The test of an effect with numerator degrees of freedom df_h (the product of the involved factors' levels each minus one) is a noncentral F with noncentrality $\lambda = N f_{\text{adj}}^2$, where N is the total sample size, $f_{\text{adj}} = f/\sqrt{1 - R^2}$, and error degrees of freedom $N - (\prod \text{levels}) - q$ for q covariates (the standard one-line ANCOVA adjustment; Maxwell, Delaney, & Kelley, 2027, Chapter 9). The covariate slopes are assumed homogeneous across cells and the covariates measured at baseline, so that adjusting does not bias the treatment effects in a randomized design.

The complete worked example for this function, a 2 x 4 x 3 ANCOVA with two baseline covariates, planned effect by effect and then analyzed with Type III sums of squares, interaction plots, and focused follow-up contrasts, is the "Power for factorial ANCOVA" vignette: `vignette("ancova_2x4x3_power", package = "DMAR")`.

Value

A `data.frame` with the per-cell and total sample sizes (or the supplied ones), the realized `actual_power`, the numerator and error degrees of freedom, the unadjusted and covariate-adjusted effect sizes (`f`, `f_adjusted`), `covariate_R2`, `n_covariates`, the noncentrality parameter, and `alpha_level`. The result carries the `dmar_ss_power` class, so `tidy` and `glance` summarize it in broom convention (the reported size is the per-cell count).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9 on designs with covariates.)

See Also

`ss_power_factorial_anova` for the no-covariate case this wraps; `ancova` and `ci_sc_ancova` for the analysis side; `vignette("ancova_2x4x3_power", package = "DMAR")` for the full worked design.

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: `power_fisher_exact()`, `ss_aipe_mixed_effects()`, `ss_power_R2()`, `ss_power_R2_sensitivity()`, `ss_power_c()`, `ss_power_c_ancova()`, `ss_power_composite_ancova()`, `ss_power_composite_ancova_2group()`, `ss_power_composite_anova()`, `ss_power_composite_factorial_ancova()`, `ss_power_composite_factorial_ancova_het()`, `ss_power_composite_factorial_anova()`, `ss_power_composite_sc`, `ss_power_contrast()`, `ss_power_equivalence_c()`, `ss_power_factorial_anova()`, `ss_power_indirect_effect()`, `ss_power_mixed_effects()`, `ss_power_one_way_anova()`, `ss_power_pcm()`, `ss_power_r()`, `ss_power_rc()`, `ss_power_reg_coef()`, `ss_power_reg_coef_sensitivity()`, `ss_power_rm_anova()`, `ss_power_sc()`, `ss_power_sem()`, `ss_power_smd()`, `ss_power_split_plot_anova()`

Examples

```
# A 2 x 4 x 3 design, planning the f = .10 main effect of the
# two-level factor A. Two baseline covariates with a modest joint
# R^2 = .25 cut the required total N by roughly a quarter:
ss_power_factorial_ancova(factor_levels = c(2, 4, 3), effect_indices = 1,
                          f = 0.10, covariate_R2 = 0,    n_covariates = 0,
                          desired_power = 0.80)
ss_power_factorial_ancova(factor_levels = c(2, 4, 3), effect_indices = 1,
                          f = 0.10, covariate_R2 = 0.25, n_covariates = 2,
                          desired_power = 0.80)

# Realized power for the three-way interaction at 6 per cell.
ss_power_factorial_ancova(factor_levels = c(2, 4, 3),
                          effect_indices = c(1, 2, 3), f = 0.15,
```

```
                          covariate_R2 = 0.25, n_covariates = 2,
                          n_per_cell = 6)
```

`ss_power_factorial_anova` *Sample Size or Power for a Factorial Between-Subjects ANOVA Effect*

Description

Determine the necessary per-cell sample size to achieve a desired level of statistical power for a single *F* test (main effect or interaction) in a between-subjects factorial ANOVA, or, given a per-cell sample size, return the realized statistical power. The function handles two-way and higher-order factorial designs.

Usage

```
ss_power_factorial_anova(
  factor_levels,
  effect_indices,
  f = NULL,
  partial_eta_squared = NULL,
  desired_power = 0.85,
  alpha_level = 0.05,
  n_per_cell = NULL
)
```

Arguments

`factor_levels`	Integer vector giving the number of levels of each factor (e.g., `c(2, 3)` for a 2 x 3 design, `c(2, 2, 4)` for a 2 x 2 x 4 design)
`effect_indices`	Integer vector identifying the factors that define the effect of interest. For example, `1` requests the main effect of the first factor; `c(1, 2)` requests the AxB two-way interaction; `c(1, 2, 3)` requests the three-way interaction
`f`	Cohen's *f* effect size for the chosen effect; supply this or `partial_eta_squared`, but not both
`partial_eta_squared`	Partial eta squared for the chosen effect; supply this or `f`
`desired_power`	Desired statistical power (default 0.85)
`alpha_level`	Type I error rate (default 0.05)
`n_per_cell`	Per-cell sample size; if specified, returns the realized power

Details

For a between-subjects factorial design, the F statistic for the chosen effect follows a noncentral F distribution under the alternative with numerator degrees of freedom $\prod_{i \in S}(k_i - 1)$ (where S is `effect_indices` and k_i is `factor_levels[i]`), denominator degrees of freedom $N - K$ (where $N = n_{cell} \prod k_i$ and $K = \prod k_i$ is the number of cells), and noncentrality parameter $\lambda = N f^2$. Cohen's f relates to partial eta squared via $f = \sqrt{\eta_p^2 / (1 - \eta_p^2)}$.

The function searches over per-cell sample sizes until power reaches `desired_power`; when `n_per_cell` is supplied it returns the realized power.

For covariates, see `ss_power_factorial_ancova`. A complete worked three-factor example (a 2 x 4 x 3 design planned effect by effect, simulated, analyzed with Type III sums of squares, plotted, and followed up with focused contrasts) is the vignette `vignette("ancova_2x4x3_power", package = "DMAR")`.

Value

A `data.frame` with rows for `necessary_n_per_cell` (or `specified_n_per_cell`), `total_N`, `effect_df`, `error_df`, `noncentrality`, and `actual_power`. The result carries the `dmar_ss_power` class, so `tidy` and `glance` summarize it in broom convention (the reported size is the per-cell count).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

`ss_power_one_way_anova`, `ss_power_c`, `ci_nc_F`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: `power_fisher_exact()`, `ss_aipe_mixed_effects()`, `ss_power_R2()`, `ss_power_R2_sensitivity()`, `ss_power_c()`, `ss_power_c_ancova()`, `ss_power_composite_ancova()`, `ss_power_composite_ancova_2group()`, `ss_power_composite_anova()`, `ss_power_composite_factorial_ancova()`, `ss_power_composite_factorial_ancova_het()`, `ss_power_composite_factorial_anova()`, `ss_power_composite_sc`, `ss_power_contrast()`, `ss_power_equivalence_c()`, `ss_power_factorial_ancova()`, `ss_power_indirect_effect()`, `ss_power_mixed_effects()`, `ss_power_one_way_anova()`, `ss_power_pcm()`, `ss_power_r()`, `ss_power_rc()`, `ss_power_reg_coef()`, `ss_power_reg_coef_sensitivity()`, `ss_power_rm_anova()`, `ss_power_sc()`, `ss_power_sem()`, `ss_power_smd()`, `ss_power_split_plot_anova()`

Examples

```
# 2 x 3 design, main effect of factor B (the 3-level factor), f = 0.25, power = .80
ss_power_factorial_anova(factor_levels = c(2, 3), effect_indices = 2,
                         f = 0.25, desired_power = 0.80)

# 2 x 2 design, AxB interaction, partial eta squared = 0.06, power = .80
ss_power_factorial_anova(factor_levels = c(2, 2), effect_indices = c(1, 2),
                         partial_eta_squared = 0.06, desired_power = 0.80)

# 2 x 2 x 3 design, three-way interaction, f = 0.20
ss_power_factorial_anova(factor_levels = c(2, 2, 3), effect_indices = c(1, 2, 3),
                         f = 0.20, desired_power = 0.80)

# Realized power for n_per_cell = 20 in a 2x3 design, AxB interaction, f = 0.25
ss_power_factorial_anova(factor_levels = c(2, 3), effect_indices = c(1, 2),
                         f = 0.25, n_per_cell = 20)
```

---

ss_power_indirect_effect

Sample Size Planning for Power for the Indirect (Mediation) Effect

---

Description

Power, or the sample size required for a desired power, for the test of the indirect effect ab in the simple mediation model, with paths specified in standardized metric (unit-variance X, M, and Y). The default test is joint significance (the indirect effect is declared when both $\hat{a}$ and $\hat{b}$ are individually significant), which tracks the resampling tests' power closely and far exceeds the Sobel test in small samples (Fritz & MacKinnon, 2007); the Sobel normal-theory test is available for comparison. This is the power counterpart of the accuracy in parameter estimation (AIPE) planner `ss_aipe_indirect_effect`, and the planning complement of the analysis function `mediate`.

Usage

```
ss_power_indirect_effect(
  a,
  b,
  c_prime = 0,
  desired_power = NULL,
  N = NULL,
  alpha_level = 0.05,
  method = c("joint_significance", "sobel")
)
```

Arguments

a	Standardized $X \rightarrow M$ path.
b	Standardized $M \rightarrow Y$ path, holding X.

c_prime	Standardized direct effect of X on Y holding M. Defaults to 0; it enters only through the residual variance of Y.
desired_power	Desired power; supply this to solve for N.
N	Total sample size; supply this to evaluate the realized power. Specify exactly one of desired_power and N. This is the total sample size.
alpha_level	Two-sided Type I error rate for each component test. Defaults to 0.05.
method	"joint_significance" (default) or "sobel".

Details

With unit-variance variables, the large-sample standard errors are $\mathrm{se}_a = \sqrt{(1-a^2)/N}$ and $\mathrm{se}_b = \sqrt{\sigma^2_{e_Y}/[N(1-a^2)]}$ with $\sigma^2_{e_Y} = 1-(b^2+c'^2+2abc')$. Because $\hat{a}$ and $\hat{b}$ are asymptotically independent in this model, the joint significance power is the product of the two component powers; the Sobel power refers ab/se_{ab} (first-order delta method, via the same variance as var_indirect_effect) to the normal. The joint-significance component powers use the exact noncentral t (with $n-2$ and $n-3$ degrees of freedom), so its only approximations are the population standard errors and component independence; the tests validate the result against raw-data simulation. A specified parameter combination must be admissible (positive residual variances), or the function stops.

Value

A tidy data.frame with necessary_N (or specified_N), actual_power (the power to detect the indirect effect, the quantity the sample size is planned against), the component powers (power_a, power_b; NA for the Sobel method), the paths (a, b, c_prime), the implied indirect_effect, and alpha_level. The method is recorded in the "method" attribute. The result carries the dmar_ss_power class, so tidy reports the sample size and the power to detect the indirect effect, and glance adds the component powers and the planning inputs.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Fritz, M. S., & MacKinnon, D. P. (2007). Required sample size to detect the mediated effect. *Psychological Science, 18*(3), 233–239. doi:10.1111/j.14679280.2007.01882.x

MacKinnon, D. P., Lockwood, C. M., Hoffman, J. M., West, S. G., & Sheets, V. (2002). A comparison of methods to test mediation and other intervening variable effects. *Psychological Methods, 7*(1), 83–104. doi:10.1037/1082989X.7.1.83

See Also

mediate to analyze the study this plans; ss_aipe_indirect_effect to plan for confidence interval width instead of detection; design_consequences for what the chosen design delivers.

Other mediation: mediate(), mediation_mbco()

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(),

ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_sc ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# Fritz and MacKinnon's (2007) running scenario (a = b = .39). The
# joint significance approximation returns necessary_N = 65; raw-data
# simulation puts the power at N = 65 nearer .77 and reaches .80 near
# N = 70, which is why Fritz and MacKinnon's simulation-based table
# reports a somewhat larger requirement.
ss_power_indirect_effect(a = .39, b = .39, desired_power = .80)

# A near-zero a path against a larger b: the weak link drives the requirement.
ss_power_indirect_effect(a = .14, b = .39, desired_power = .80)

# Realized power at a given N, and the Sobel comparison (always lower).
ss_power_indirect_effect(a = .39, b = .39, N = 75)
ss_power_indirect_effect(a = .39, b = .39, N = 75, method = "sobel")
```

---

ss_power_mixed_effects

Sample Size or Power for a Treatment Effect in a Two-Level Mixed-Effects Model

---

Description

Determine the necessary number of level-2 units per arm to achieve a desired level of statistical power for a treatment-versus-control comparison in a two-level mixed-effects model with a random intercept (e.g., individuals nested within clusters in a cluster-randomized trial, or repeated measurements nested within subjects in a person-randomized longitudinal study). Alternatively, given a number of level-2 units per arm, return the realized statistical power.

Usage

```
ss_power_mixed_effects(
  d,
  n,
  rho,
  J = NULL,
  desired_power = 0.85,
  alpha_level = 0.05,
  directional = FALSE
)
```

Arguments

d	Standardized treatment effect, defined as the population mean difference divided by the population standard deviation of the level-1 outcome
n	Number of level-1 units per level-2 unit (e.g., individuals per cluster, or measurements per subject); assumed equal across level-2 units
rho	Intra-class correlation (the proportion of total outcome variance attributable to differences between level-2 units); must be in [0, 1)
J	Number of level-2 units per arm (i.e., J treatment clusters and J control clusters); if specified, returns the realized power
desired_power	Desired statistical power (default 0.85)
alpha_level	Type I error rate (default 0.05)
directional	Logical: TRUE for a one-sided test (in the same sign as d), FALSE (default) for a two-sided test

Details

This function computes power for the fixed treatment effect at the higher level of a two-level mixed-effects model with random intercept,

$$y_{ij} = \beta_0 + \beta_1 T_j + u_j + \epsilon_{ij},$$

where $u_j \sim N(0, \sigma_u^2)$ is the level-2 random intercept and $\epsilon_{ij} \sim N(0, \sigma_e^2)$ is the level-1 residual. The treatment indicator T_j varies between level-2 units (i.e., entire clusters or entire subjects are assigned to treatment or control). The intra-class correlation is $\rho = \sigma_u^2/(\sigma_u^2 + \sigma_e^2)$ and the total outcome variance is $\sigma_y^2 = \sigma_u^2 + \sigma_e^2$.

The standard error of the estimated treatment effect is $SE(\hat{\beta}_1) = \sigma_y \sqrt{2(1 + (n-1)\rho)/(Jn)}$, giving a noncentrality parameter of

$$\lambda = d\sqrt{Jn/(2(1 + (n-1)\rho))}$$

under a two-sample *t*-test with $2J - 2$ degrees of freedom.

The factor $1 + (n-1)\rho$ is the design effect: as the within-cluster correlation grows, the effective information per level-2 unit shrinks, so more level-2 units are needed for a given level of power.

Value

A data.frame with rows for necessary_J_per_arm (or specified_J_per_arm), total_N, noncentrality, and actual_power. The result carries the dmar_ss_power class, so tidy and glance summarize it in broom convention (the reported size is the number of clusters per arm).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Raudenbush, S. W. (1997). Statistical analysis and optimal design for cluster randomized trials. *Psychological Methods, 2*, 173–185. doi:10.1037/1082989X.2.2.173

See Also

ss_power_split_plot_anova, ss_power_smd, ss_power_pcm

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_s ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Other mixed models: R2_mixed_effects(), R2_mixed_effects_decomposition(), icc_lmer(), manova_split_plot(), mixed_anova(), ss_aipe_mixed_effects(), ss_aipe_mixed_effects_sensitivity(), ss_power_split_plot_anova()

Examples

```
# 30 individuals per cluster, ICC = 0.05, standardized effect d = 0.30, power = .80
ss_power_mixed_effects(d = 0.30, n = 30, rho = 0.05, desired_power = 0.80)

# Same effect but with much higher ICC (e.g., schools or therapists)
ss_power_mixed_effects(d = 0.30, n = 30, rho = 0.20, desired_power = 0.80)

# Realized power with 25 level-2 units per arm
ss_power_mixed_effects(d = 0.30, n = 30, rho = 0.05, J = 25)

# directional test
ss_power_mixed_effects(d = 0.30, n = 30, rho = 0.05, desired_power = 0.80,
                       directional = TRUE)
```

---

ss_power_one_way_anova

Sample Size or Power for a One-Way Between-Subjects ANOVA Omnibus F *Test*

Description

Determine the necessary total sample size to achieve a desired level of statistical power for the omnibus F test in a one-way between-subjects analysis of variance, or, given a total sample size, return the realized statistical power.

Usage

```
ss_power_one_way_anova(
  a,
  f = NULL,
  eta_squared = NULL,
  desired_power = 0.85,
  alpha_level = 0.05,
  N = NULL
)
```

Arguments

a	Number of groups (levels of the between-subjects factor)
f	Cohen's f effect size (the population value); supply this or eta_squared, but not both
eta_squared	Population eta squared (proportion of total variance accounted for by group membership); supply this or f
desired_power	Desired statistical power (default 0.85)
alpha_level	Type I error rate (default 0.05)
N	Total sample size; if specified, the function returns the realized power (the ss_power_* family is not uniform here: ss_power_contrast takes a *per-group* size)

Details

Under the alternative hypothesis, the omnibus F statistic follows a noncentral F distribution with numerator df $a-1$, denominator df $N-a$, and noncentrality parameter $\lambda = N f^2$. Cohen's f relates to eta squared via $f = \sqrt{\eta^2/(1-\eta^2)}$.

The function searches over total sample sizes N (treating per-group N/a as balanced) until power first reaches desired_power. When N is supplied it instead reports the realized power at that N.

Value

A data.frame. When a sample size is being planned (N not supplied) the rows are necessary_N, n_per_group, a, noncentrality, and actual_power; the search constructs the total as a balanced design, so n_per_group is a whole-number per-group count. When N is supplied, power is evaluated at that total N directly and the rows are specified_N, a, noncentrality, and actual_power (no n_per_group row, since balance is not assumed). The result carries the dmar_ss_power class, so tidy and glance summarize it in broom convention; the summarized sample size is the total N.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

ss_power_factorial_anova, ss_power_c, ss_power_sc, ci_nc_F

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_sc ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# Three groups, f = 0.25, power = .80
ss_power_one_way_anova(a = 3, f = 0.25, desired_power = 0.80)

# Same effect specified via eta squared
ss_power_one_way_anova(a = 3, eta_squared = 0.0588, desired_power = 0.80)

# Realized power at N = 60 across 3 groups
ss_power_one_way_anova(a = 3, f = 0.25, N = 60)
```

---

ss_power_pcm *Sample Size Planning for Power for Polynomial Change Models*

Description

Returns power given the sample size, or sample size given the desired power, for the group difference in a polynomial change coefficient (a flat-line intercept, a linear slope, a quadratic acceleration, or any higher-order trend) in a two-group longitudinal design, following Raudenbush and Liu (2001). The trend whose group difference is tested is selected with `trend`; `trend = "linear"` (the default) reproduces the straight-line case.

Usage

```
ss_power_pcm(
  beta,
  tau,
  level_1_variance,
  frequency,
  duration,
  desired_power = NULL,
  N = NULL,
  alpha_level = 0.05,
  standardized = TRUE,
  directional = FALSE,
  trend = "linear"
)
```

Arguments

beta	The level two regression coefficient for the group by time interaction in the polynomial change coefficient selected by `trend` (linear by default), where the grouping variable is coded -.5 and .5 for the two groups. When `standardized = TRUE` (the default) this is the standardized change difference (see `standardized`).
tau	The true between-subject variance of the individuals' change coefficient for the trend selected by `trend` (the variance of the slopes for `trend = "linear"`).
level_1_variance	Level one (within-subject) error variance
frequency	Number of measurements per unit of time, where the unit is the one in which `duration` is expressed. It need not be a whole number; for example, `frequency = 0.5` means one measurement every two time units. Together with `duration` it fixes the number of equally spaced measurement occasions, $M = f \times D + 1$, with f the frequency and D the duration.
duration	Length of the study in the chosen time unit (for example, years, grades, or hours). Measurements are taken at $M = f \times D + 1$ equally spaced occasions spanning time 0 to `duration`.
desired_power	Desired power
N	Total sample size (one-half in each of the two groups)
alpha_level	Type I error rate
standardized	The standardized change difference is the unstandardized change difference divided by the square root of `tau`, the between-subject variance of the change coefficient. `TRUE` (the default) treats `beta` as already standardized; `FALSE` treats it as the raw (unstandardized) change difference.
directional	Should a one (`TRUE`) or two (`FALSE`) tailed test be performed.
trend	The polynomial change coefficient whose group difference is tested, given either as a name (`"intercept"`, `"linear"`, `"quadratic"`, `"cubic"`, `"quartic"`, ...) or as a non-negative integer order (`0` = intercept / flat line, `1` = linear, `2` = quadratic, ...). Defaults to `"linear"`. The design must supply at least $p + 1$ measurement occasions to estimate a degree-p trend.

Details

The two groups each contain $N/2$ subjects measured on $M = f \times D + 1$ equally spaced occasions. Each subject's degree-p polynomial change coefficient is estimated within subject; the test compares the two group means of that coefficient. The change coefficient is taken in the derivative-scaled metric ($p!$ times the leading coefficient of t^p), the metric in which the Raudenbush and Liu (2001) constants apply.

The within-subject sampling variance of the estimated coefficient is (Raudenbush & Liu, 2001, p. 392)

$$V = \sigma_e^2 \, f^{2p} \, \frac{(M-p-1)!}{K_p \, (M+p)!}, \qquad \frac{1}{K_p} = \frac{(2p)! \, (2p+1)!}{(p!)^2},$$

so that $1/K_p = 1, 12, 720, 100800, \ldots$ for $p = 0, 1, 2, 3, \ldots$. This V equals $(p!)^2$ times the variance of the ordinary least squares estimate of the coefficient of t^p, reduces to σ_e^2/M at $p = 0$ and to $12\,\sigma_e^2 f^2/[M(M^2-1)]$ at $p = 1$. The slope reliability is $\tau/(\tau+V)$ (their Equation 15), the variance of the between-group difference is $4(\tau+V)/N$ (their Equation 10), and the *t* test has $N-2$ degrees of freedom and noncentrality $\sqrt{N\,\beta^2\,[\tau/(\tau+V)]/4}$ (their Equations 12 and 14). The linear case reproduces the National Youth Survey benchmark in their Tables 1 and 2; the general-p formula has been checked against the exact $(X'X)^{-1}$ variance and against an end-to-end Monte Carlo power study for the quadratic trend.

Value

A `data.frame` (class `dmar_tbl`) with one row per reported quantity in a `term` / `value` layout: the per-group size per group (`necessary_n_per_group`, or `specified_n_per_group` when `N` is supplied) and the total (`total_N`); the achieved power (`actual_power`); the measurement schedule (`freq`, `duration`, `measurement_occasions`); the polynomial order of the tested change coefficient (`polynomial_order`, 0 = intercept, 1 = linear, 2 = quadratic, ...); the unstandardized and standardized change difference (`unstd_coefficient`, `std_coefficient`); the level one error variance (`l1_error_var`); the true and error variance of the change coefficient (`true_var_of_slopes`, `error_var_of_slopes`, whose names retain "slopes" from the linear case); the change-coefficient reliability (`reliability`); and the noncentrality parameter of the *t* test (`noncentral_t_parm`).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Kelley, K., & Rausch, J. R. (2011). Sample size planning for longitudinal models: Accuracy in parameter estimation for polynomial change parameters. *Psychological Methods, 16*(4), 391–405. doi:10.1037/a0023352

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapters 11, 15.)

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

Raudenbush, S. W., & Liu, X.-F. (2001). Effects of study duration, frequency of observation, and sample size on power in studies of group differences in polynomial change. *Psychological Methods, 6*(4), 387–401. doi:10.1037/1082989X.6.4.387

See Also

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_sc, ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# The examples reproduce the National Youth Survey illustration of
# Raudenbush and Liu (2001, p. 393). One observation per year
# (frequency = 1) over a four-year study (duration = 4) gives
# M = frequency * duration + 1 = 5 equally spaced occasions. The
# standardized slope difference is -0.40, the slope variance is
# tau = 0.003, and the level-one error variance is 0.0262, so the slope
# reliability is 0.53.

# (1) Power at a given sample size. With N = 238 (119 per group) the
#     design has power 0.61 (Raudenbush and Liu, 2001, Table 1, D = 4,
#     f = 1).
ss_power_pcm(beta = -.4, tau = .003, level_1_variance = .0262,
             frequency = 1, duration = 4, N = 238)

# (2) Sample size for a target power. Solving the same design for 0.80
#     power returns N = 370 (185 per group), between their Table 2 cells
#     N = 300 (power 0.71) and N = 400 (power 0.83).
ss_power_pcm(beta = -.4, tau = .003, level_1_variance = .0262,
             frequency = 1, duration = 4, desired_power = .80)

# (3) Unstandardized slope. The unstandardized slope difference is
#     beta * sqrt(tau) = -0.40 * sqrt(0.003) = -0.0219. Passing it with
#     standardized = FALSE reproduces the power of 0.61 from example (1).
ss_power_pcm(beta = -.0219, tau = .003, level_1_variance = .0262,
             frequency = 1, duration = 4, N = 238, standardized = FALSE)

# (4) Longer study, same number of occasions. Doubling the duration to
#     D = 8 while keeping M = 5 (so frequency = 0.5, one observation every
#     two years) raises the slope reliability to 0.82 and power to about
```

```
#     0.80. Spreading the same five occasions over a longer span sharply
#     increases power (Raudenbush & Liu, 2001, p. 393).
ss_power_pcm(beta = -.4, tau = .003, level_1_variance = .0262,
             frequency = .5, duration = 8, N = 238)

# (5) More frequent sampling over a shorter span, same occasions. Halving
#     the duration to D = 2 while keeping M = 5 (so frequency = 2) drops
#     the slope reliability to 0.22 and power to about 0.31. Sampling more
#     often over a shorter study does little for power (Raudenbush & Liu,
#     2001, p. 393).
ss_power_pcm(beta = -.4, tau = .003, level_1_variance = .0262,
             frequency = 2, duration = 2, N = 238)

# (6) One-sided test. A directional test of the base design places the
#     whole Type I error rate in the predicted tail, raising power from
#     0.61 to about 0.73.
ss_power_pcm(beta = -.4, tau = .003, level_1_variance = .0262,
             frequency = 1, duration = 4, N = 238, directional = TRUE)

# (7) A higher-order trend. The same machinery plans power for the group
#     difference in any polynomial change coefficient. Here the target is
#     the quadratic trend (curvature / acceleration): with eight occasions
#     (frequency = 1, duration = 7), a between-subject quadratic-coefficient
#     variance tau = 0.002, level-one error variance 0.05, and a
#     standardized quadratic difference of 0.45, the design is planned for
#     0.80 power. A quadratic trend needs at least three occasions; a cubic
#     at least four (trend = "cubic" or trend = 3), and so on.
ss_power_pcm(beta = 0.45, tau = 0.002, level_1_variance = 0.05,
             frequency = 1, duration = 7, desired_power = .80,
             trend = "quadratic")
```

`ss_power_r` *Sample Size or Power for a Pearson Correlation Coefficient (Fisher Z Transformation)*

Description

Determine the necessary sample size to achieve a desired level of statistical power for the test of a Pearson correlation against a null value (typically zero), or, given a sample size, return the realized statistical power. The computation uses the Fisher's Z transformation, which has a near-normal sampling distribution.

Usage

```
ss_power_r(
  rho,
  rho_0 = 0,
  desired_power = 0.85,
```

```
  alpha_level = 0.05,
  N = NULL,
  directional = FALSE
)
```

Arguments

rho	The population correlation coefficient under the alternative hypothesis
rho_0	The null hypothesis value of the correlation (default 0)
desired_power	Desired statistical power (default 0.85)
alpha_level	Type I error rate (default 0.05)
N	Sample size (*number of pairs*); if specified, returns the realized power (the ss_power_* family is not uniform here: ss_power_contrast takes a *per-group* size)
directional	Logical: TRUE for a one-sided test (in the same direction as the difference rho - rho_0), FALSE (default) for a two-sided test

Details

Under the alternative the Fisher-transformed correlation $Z_r = \tanh^{-1}(r)$ is approximately normal with mean $Z_\rho = \tanh^{-1}(\rho)$ and variance $1/(N-3)$. Power is computed from this normal approximation.

For sample size, a closed-form expression is used as the starting point,

$$N = ((z_\alpha + z_\beta)/(Z_\rho - Z_{\rho_0}))^2 + 3,$$

which is then verified iteratively to ensure power exactly meets or exceeds desired_power. The search is bounded at $N = 10^7$. When rho and rho_0 are so close that desired_power is unreachable within that bound, the function stops with an error rather than searching indefinitely.

Value

A data.frame with rows for necessary_N (or specified_N), actual_power, rho, and rho_0.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Fisher, R. A. (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron, 1*, 3–32.

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on the one-way ANOVA and Chapter 4 on contrasts.)

See Also

ci_r, convert_r_Z, convert_Z_r

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_s, ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# Population r = 0.30, null r = 0, desired power = .80, two-sided
ss_power_r(rho = 0.30, desired_power = 0.80)

# Same with a directional alternative
ss_power_r(rho = 0.30, desired_power = 0.80, directional = TRUE)

# Realized power for N = 100 pairs
ss_power_r(rho = 0.30, N = 100)

# Test against a non-zero null (rho_0 = 0.20) -- looking for evidence rho > 0.20
ss_power_r(rho = 0.40, rho_0 = 0.20, desired_power = 0.80, directional = TRUE)
```

---

ss_power_R2 — *Plan Sample Size to Make the Test of the Squared Multiple Correlation Coefficient Sufficiently Powerful*

Description

Determine the necessary sample size for the omnibus test of the squared multiple correlation coefficient (R^2), or the realized statistical power given a specified sample size, under either fixed or random predictors. The fixed-predictors path uses Cohen's (1988) noncentral F formulation; the random-predictors path uses the Lee (1971) two-moment approximation to the sampling distribution of the sample R^2 under joint multivariate normality.

Usage

```
ss_power_R2(
  population_R2 = NULL,
  alpha_level = 0.05,
  desired_power = 0.85,
  p,
  specified_N = NULL,
  cohen_f2 = NULL,
  null_R2 = 0,
  random_predictors = TRUE,
  print_progress = FALSE,
  ...
)
```

Arguments

population_R2	Population squared multiple correlation coefficient
alpha_level	Type I error rate
desired_power	Desired degree of statistical power
p	The number of predictor variables
specified_N	The sample size used to calculate power (rather than determine necessary sample size). This is the *total* sample size across all groups or observations.
cohen_f2	Cohen's (1988) effect size for multiple regression: population_R2/(1-population_R2)
null_R2	Value of the null hypothesis that the squared multiple correlation will be evaluated against (this will typically be zero)
random_predictors	Whether the predictor variables are treated as random (TRUE, the default) or fixed (FALSE). See Details.
print_progress	If the progress of the iterative procedure is printed to the screen as the iterations are occurring
...	Possible additional parameters for internal functions

Details

Determine the necessary sample size given a particular population_R2, alpha_level, p, and desired_power. Alternatively, given population_R2, alpha_level, p, and specified_N, the function can be used to determine the statistical power.

Fixed vs.\ random predictors. The two regression models give *different* sampling distributions for the omnibus F-statistic, and so different power. Under fixed predictors the design matrix is treated as constant in hypothetical replications of the study, and F follows a noncentral F with p and $N - p - 1$ degrees of freedom and noncentrality $\lambda = N \cdot f^2$, where $f^2 = \rho^2/(1 - \rho^2)$ (Cohen, 1988). Under random predictors the design matrix is itself a draw from a joint multivariate normal distribution, and the unconditional distribution of the sample R^2 is given by Lee (1971); ss_power_R2() uses Lee's two-moment Patnaik (1949) approximation to that distribution, the same approximation ci_R2() uses for random-predictor confidence intervals. Gatsonis and Sampson

(1989) document the comparison and show that Cohen's fixed-predictor formula tends to over-state power (and so under-state required N) relative to the random model; the discrepancy is modest at moderate to large N but non-trivial for small N with moderate-to-large effects. In the behavioral, educational, and social sciences predictor variables are almost always random, so the default is `random_predictors = TRUE`; pass `random_predictors = FALSE` for designs in which the predictor variables are fixed by design (for example, planned dosing levels).

Value

A `data.frame` with columns `term` and `value`. For an N search the rows are `necessary_N`, `actual_power`, `noncentral_f_parm` (only meaningful for `random_predictors = FALSE`; `NA` otherwise), and `effect_size` (Cohen's f^2). For a power-at-specified-`N` computation the first row is `specified_N` instead of `necessary_N`.

Note

When determining sample size for a desired degree of power, there will always be a slightly larger degree of actual power. This is the case because the algorithm employed determines sample size until the actual power is no less than the desired power (given sample size is a whole number power will almost certainly not be exactly the specified value). This is the same as other statistical power procedures that return whole numbers for necessary sample size.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Gatsonis, C., & Sampson, A. R. (1989). Multiple correlation: Exact power and sample size calculations. *Psychological Bulletin, 106*(3), 516–524.

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*(4), 524–555. doi:10.1080/00273170802490632

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Lee, Y. S. (1971). Some results on the sampling distribution of the multiple correlation coefficient. *Journal of the Royal Statistical Society, Series B, 33*(1), 117–130.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on R^2 as a model comparison effect size.)

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

Patnaik, P. B. (1949). The non-central χ^2- and *F*-distributions and their applications. *Biometrika, 36*(1–2), 202–232. doi:10.1093/biomet/36.12.202

Anderson, S. F., Kelley, K., & Maxwell, S. E. (2017). Sample-size planning for more accurate statistical power: A method adjusting sample effect sizes for publication bias and uncertainty. *Psychological Science, 28*(11), 1547–1562. doi:10.1177/0956797617723724

See Also

ss_aipe_R2, ss_power_R2_sensitivity, ss_power_reg_coef, ci_nc_F, ci_R2

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova
ss_power_composite_factorial_anova(), ss_power_composite_sem(), ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# Random predictors (default; appropriate for most behavioral / social
# science applications).
ss_power_R2(population_R2 = .5, alpha_level = .05, desired_power = .85, p = 5)

# Fixed predictors (Cohen 1988): predictor variables fixed by design.
ss_power_R2(population_R2 = .5, alpha_level = .05, desired_power = .85,
            p = 5, random_predictors = FALSE)

# Effect size input (Cohen's f^2).
ss_power_R2(cohen_f2 = 1, alpha_level = .05, desired_power = .85, p = 5)

# Realized power at a specified N.
ss_power_R2(population_R2 = .5, specified_N = 15, alpha_level = .05,
            desired_power = .85, p = 5)
```

---

ss_power_R2_sensitivity

Sensitivity Analysis for Sample Size Planning to Make the Omnibus Test of R^2 Sufficiently Powerful

---

Description

Monte Carlo sensitivity analysis for the power of the omnibus F-test of the squared multiple correlation coefficient (R^2). Given an `estimated_R2` used for sample size planning and a `true_R2` that actually obtains in the population (the two need not agree), the function draws `G` replications, fits the regression, compares F to its critical value, and reports the realized empirical power and a summary of the realized R^2 and F distributions. The simulation honors the same `random_predictors` / `generate_random_predictors` crossing as `ss_aipe_R2_sensitivity`, so the user can examine the effect of planning under one regression model (fixed or random predictors) but actually realizing the other.

Usage

```
ss_power_R2_sensitivity(
  true_R2 = NULL,
  estimated_R2 = NULL,
  desired_power = 0.85,
  p = NULL,
  alpha_level = 0.05,
  random_predictors = TRUE,
  specified_N = NULL,
  generate_random_predictors = TRUE,
  rho_yx = 0.3,
  rho_xx = 0.3,
  G = 10000,
  print_iter = TRUE,
  filename = NULL
)
```

Arguments

`true_R2`	Value of the population squared multiple correlation coefficient
`estimated_R2`	Value of the squared multiple correlation coefficient used for sample size planning. Either `estimated_R2` or `specified_N` must be supplied (not both).
`desired_power`	Desired degree of statistical power used for planning
`p`	Number of predictors
`alpha_level`	Type I error rate
`random_predictors`	Whether the sample size planning step treats predictors as random (`TRUE`, the default) or fixed (`FALSE`)
`specified_N`	Sample size at which the realized power should be computed; alternative to specifying `estimated_R2`
`generate_random_predictors`	Whether the internal simulation should generate predictors as random (`TRUE`, the default) or fixed (`FALSE`)
`rho_yx`	Correlation between the dependent variable (Y) and each of the X variables

rho_xx	Correlation among the X variables (off-diagonal of the predictor correlation matrix)
G	Number of Monte Carlo replications
print_iter	Whether to print the iteration number during the simulation
filename	Optional path of a CSV file to receive the per-replication results (the observed R^2 and F statistic), overwriting any file already at that path; the default NULL writes nothing, and a throwaway run that wants the file should point it at tempfile(fileext = ".csv").

Details

When estimated_R2 equals true_R2, the function performs a straight Monte Carlo evaluation of the planning procedure (no misspecification). Pass specified_N to evaluate realized power at a specified sample size; in that case estimated_R2 must not be supplied. The crossing of random_predictors (used in planning) with generate_random_predictors (used in the simulation) lets the user inspect the consequences of planning under one regression model but realizing the other. See Gatsonis and Sampson (1989) for the comparison of fixed and random predictor power for the omnibus test.

Value

A data.frame with columns term and value summarizing the Monte Carlo sensitivity analysis across G replications. The term entries are: total_N (the sample size evaluated), empirical_power (the proportion of replications on which F exceeded the critical value), analytic_power (computed from ss_power_R2 under the same model as planning), mean_R2 / median_R2 / sd_R2 and mean_F / median_F / sd_F (summaries of the realized R^2 and F), F_crit (the critical value), and the input echoes p, true_R2, estimated_R2 and desired_power (both NA when specified_N was supplied instead), and alpha_level. The result carries the dmar_ss_power_sensitivity class, so tidy reports the planned sample size beside the empirical and analytic power, and glance adds the simulated R^2 and F distribution beside the echoed inputs.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Gatsonis, C., & Sampson, A. R. (1989). Multiple correlation: Exact power and sample size calculations. *Psychological Bulletin, 106*(3), 516–524.

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Lee, Y. S. (1971). Some results on the sampling distribution of the multiple correlation coefficient. *Journal of the Royal Statistical Society, Series B, 33*(1), 117–130.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on R^2 as a model comparison effect size.)

See Also

ss_power_R2, ss_aipe_R2_sensitivity, ci_R2

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova ss_power_composite_factorial_anova(), ss_power_composite_sem(), ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
set.seed(113)
# Realized power when planning under the fixed-predictor model but the
# data are actually generated with random predictors. G = 200 keeps the
# example quick; a reported analysis deserves the default G of 10000.
ss_power_R2_sensitivity(true_R2 = 0.30, estimated_R2 = 0.30,
                        desired_power = 0.80, p = 5,
                        random_predictors = FALSE,
                        generate_random_predictors = TRUE,
                        G = 200, print_iter = FALSE)
```

---

ss_power_rc *Sample Size for a Targeted Regression Coefficient*

Description

Determine the necessary sample size for a targeted regression coefficient or determine the degree of power given a specified sample size.

Usage

```
ss_power_rc(
  rho2_Y_X = NULL,
  rho2_Y_X_without_j = NULL,
  p = NULL,
  desired_power = 0.85,
  alpha_level = 0.05,
  directional = FALSE,
```

```
  beta_j = NULL,
  sigma_X = NULL,
  sigma_Y = NULL,
  Rho2_j_X_without_j = NULL,
  rho_XX = NULL,
  rho_YX = NULL,
  which_predictor = NULL,
  cohen_f2 = NULL,
  specified_N = NULL,
  print_progress = FALSE
)
```

Arguments

rho2_Y_X	Population squared multiple correlation coefficient predicting the dependent variable (i.e., Y) from the p predictor variables (i.e., the X variables)
rho2_Y_X_without_j	Population squared multiple correlation coefficient predicting the dependent variable (i.e., Y) from the p-1 predictor variables, where the one not used is the predictor of interest
p	Number of predictor variables
desired_power	Desired degree of statistical power for the test of targeted regression coefficient
alpha_level	Type I error rate
directional	Whether or not a direction or a nondirectional test is to be used (usually directional=FALSE)
beta_j	Population value of the regression coefficient for the predictor of interest
sigma_X	Population standard deviation for the predictor variable of interest
sigma_Y	Population standard deviation for the outcome variable
Rho2_j_X_without_j	Population squared multiple correlation coefficient predicting the predictor variable of interest from the remaining p-1 predictor variables
rho_XX	Population correlation matrix for the p predictor variables
rho_YX	Population vector of correlation coefficient between the p predictor variables and the criterion variable
which_predictor	Identifies the predictor of interest when rho_XX and rho_YX are specified
cohen_f2	Cohen's (1988) definition for an effect size for a targeted regression coefficient: (rho2_Y_X - rho2_Y_X_without_j) / (1 - rho2_Y_X)
specified_N	Sample size for which power should be evaluated. This is the *total* sample size.
print_progress	If the progress of the iterative procedure is printed to the screen as the iterations are occurring

Details

Determines the necessary sample size given a desired level of statistical power. Alternatively, determines the statistical power for a given a specified sample size. There are a number of ways that

the specification regarding the size of the regression coefficient can be entered. The most basic, and often the simplest, is to specify `rho2_Y_X` and `rho2_Y_X_without_j`. See the examples section for several options.

Value

A tidy `data.frame` with a `term` column and a numeric `value` column, forwarded unchanged from `ss_power_reg_coef`: rows for `necessary_N` (the necessary total sample size) or `specified_N` (when `specified_N` is supplied), `actual_power`, `noncentral_t_parm` (the noncentrality of the t distribution), and `effect_size` (the square root of `cohen_f2`, since `cohen_f2` is the effect size on the F scale). The result carries the `dmar_ss_power` class, so `tidy` and `glance` summarize it in broom convention.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Anderson, S. F., Kelley, K., & Maxwell, S. E. (2017). Sample-size planning for more accurate statistical power: A method adjusting sample effect sizes for publication bias and uncertainty. *Psychological Science, 28*(11), 1547–1562. doi:10.1177/0956797617723724

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Maxwell, S. E. (2000). Sample size and multiple regression analysis. *Psychological Methods, 5*(4), 434–458. doi:10.1037/1082989X.5.4.434

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means and Chapter 6 on trend analysis.)

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

See Also

`ss_aipe_reg_coef`, `ss_power_R2`, `ci_nc_F`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: `power_fisher_exact()`, `ss_aipe_mixed_effects()`, `ss_power_R2()`, `ss_power_R2_sensitivity()`, `ss_power_c()`, `ss_power_c_ancova()`, `ss_power_composite_ancova()`, `ss_power_composite_ancova_2group()`, `ss_power_composite_anova()`, `ss_power_composite_factorial_ancova()`,

ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_s
ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(),
ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(),
ss_power_r(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(),
ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
Cor.Mat <- rbind(
  c(1.00, 0.53, 0.58, 0.60, 0.46, 0.66),
  c(0.53, 1.00, 0.35, 0.07, 0.14, 0.43),
  c(0.58, 0.35, 1.00, 0.18, 0.29, 0.50),
  c(0.60, 0.07, 0.18, 1.00, 0.30, 0.26),
  c(0.46, 0.14, 0.29, 0.30, 1.00, 0.30),
  c(0.66, 0.43, 0.50, 0.26, 0.30, 1.00))

rho_XX <- Cor.Mat[2:6, 2:6]
rho_YX <- Cor.Mat[1, 2:6]

# Method 1
ss_power_rc(rho2_Y_X = 0.7826786, rho2_Y_X_without_j = 0.7363697, p = 5,
            alpha_level = .05, directional = FALSE, desired_power = .80)

# Method 2
ss_power_rc(alpha_level = .05, rho_XX = rho_XX, rho_YX = rho_YX, which_predictor = 5,
            directional = FALSE, desired_power = .80)

# Method 3
# Here, beta_j is the standardized regression coefficient. Had beta_j
# been the unstandardized regression coefficient, sigma_X and sigma_Y
# would have been the standard deviation for the X variable of interest
# and Y, respectively.
ss_power_rc(rho2_Y_X = 0.7826786, Rho2_j_X_without_j = 0.3652136, beta_j = 0.2700964, p = 5,
            alpha_level = .05, sigma_X = 1, sigma_Y = 1,
            directional = FALSE, desired_power = .80)
# Method 4
ss_power_rc(alpha_level = .05, cohen_f2 = 0.2130898, p = 5,
            directional = FALSE, desired_power = .80)
# Power given a specified N and squared multiple correlation coefficients.
ss_power_rc(rho2_Y_X = 0.7826786, rho2_Y_X_without_j = 0.7363697, specified_N = 25, p = 5,
            alpha_level = .05, directional = FALSE)

# Power given a specified N and effect size.
ss_power_rc(alpha_level = .05, cohen_f2 = 0.2130898, p = 5, specified_N = 25, directional = FALSE)

# Reproducing Maxwell's (2000, p. 445) Example
Cor.Mat.Maxwell <- rbind(
    c(1.00, 0.35,  0.20, 0.20, 0.20, 0.20),
    c(0.35, 1.00,  0.40, 0.40, 0.40, 0.40),
    c(0.20, 0.40,  1.00, 0.45, 0.45, 0.45),
    c(0.20, 0.40,  0.45, 1.00, 0.45, 0.45),
    c(0.20, 0.40,  0.45, 0.45, 1.00, 0.45),
```

```
    c(0.20, 0.40,  0.45, 0.45, 0.45, 1.00)
)

RHO.XX.Maxwell <- Cor.Mat.Maxwell[2:6, 2:6]
Rho.YX.Maxwell <- Cor.Mat.Maxwell[1, 2:6]
R2.Maxwell <- Rho.YX.Maxwell %*% solve(RHO.XX.Maxwell) %*% Rho.YX.Maxwell

RHO.XX.Maxwell.no.1 <- Cor.Mat.Maxwell[3:6, 3:6]
Rho.YX.Maxwell.no.1 <- Cor.Mat.Maxwell[1, 3:6]
R2.Maxwell.no.1 <- Rho.YX.Maxwell.no.1 %*% solve(RHO.XX.Maxwell.no.1) %*% Rho.YX.Maxwell.no.1


# Note that Maxwell arrives at N=113, whereas this procedure arrives at 111.
# This seems to be the case becuase of rounding error in calculations
# and tables (Cohen, 1988) used. The present procedure is correct and
# contains no rounding error in the application of the method.
ss_power_rc(rho2_Y_X = R2.Maxwell, rho2_Y_X_without_j = R2.Maxwell.no.1, p = 5,
            alpha_level = .05, directional = FALSE, desired_power = .80)
```

`ss_power_reg_coef` *Sample Size for a Targeted Regression Coefficient*

Description

Determine the necessary sample size for a targeted regression coefficient or determine the degree of power given a specified sample size

Usage

```
ss_power_reg_coef(
  rho2_Y_X = NULL,
  rho2_Y_X_without_j = NULL,
  p = NULL,
  desired_power = 0.85,
  alpha_level = 0.05,
  directional = FALSE,
  beta_j = NULL,
  sigma_X = NULL,
  sigma_Y = NULL,
  rho2_j_X_without_j = NULL,
  rho_XX = NULL,
  rho_YX = NULL,
  which_predictor = NULL,
  cohen_f2 = NULL,
  specified_N = NULL,
  print_progress = FALSE
)
```

Arguments

rho2_Y_X	Population squared multiple correlation coefficient predicting the dependent variable (i.e., *Y*) from the p predictor variables (i.e., the *X* variables)
rho2_Y_X_without_j	Population squared multiple correlation coefficient predicting the dependent variable (i.e., *Y*) from the p-1 predictor variables, where the one not used is the predictor of interest
p	Number of predictor variables
desired_power	Desired degree of statistical power for the test of targeted regression coefficient
alpha_level	Type I error rate
directional	Whether or not a direction or a nondirectional test is to be used (usually directional=FALSE)
beta_j	Population value of the regression coefficient for the predictor of interest
sigma_X	Population standard deviation for the predictor variable of interest
sigma_Y	Population standard deviation for the outcome variable
rho2_j_X_without_j	Population squared multiple correlation coefficient predicting the predictor variable of interest from the remaining p-1 predictor variables
rho_XX	Population correlation matrix for the p predictor variables
rho_YX	Population vector of correlation coefficient between the p predictor variables and the criterion variable
which_predictor	Identifies the predictor of interest when rho_XX and rho_YX are specified
cohen_f2	Cohen's (1988) definition for an effect size for a targeted regression coefficient: (rho2_Y_X-rho2_Y_X_without_j)/(1-rho2_Y_X)
specified_N	Sample size for which power should be evaluated. This is the *total* sample size.
print_progress	If the progress of the iterative procedure is printed to the screen as the iterations are occurring

Details

Determines the necessary sample size given a desired level of statistical power. Alternatively, determines the statistical power for a given a specified sample size. There are a number of ways that the specification regarding the size of the regression coefficient can be entered. The most basic, and often the simplest, is to specify rho2_Y_X and rho2_Y_X_without_j. See the examples section for several options.

Power is computed from a noncentral t distribution with noncentrality $\sqrt{N}\,f$, which treats the predictors as fixed (their values held constant across hypothetical replications). This is the standard fixed-predictor power analysis; under random predictors, where the predictor values themselves vary from sample to sample, the sample size required for a given level of power is somewhat larger.

Value

ss	Either the necessary sample size or the specified sample size, depending if one is interested in determining the necessary sample size given a desired degree of statistical power or if one is interested in the determining the value of statistical power given a specified sample size, respectively
actual_power	Actual power of the situation described
noncentral_t_parm	Value of the noncentral distribution for the appropriate *t*-distribution
effect_size	Effect size for the noncentral *t*-distribution; this is the square root of cohen_f2, because cohen_f2 is the effect size using an *F*-distribution

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Maxwell, S. E. (2000). Sample size and multiple regression analysis. *Psychological Methods, 5*(4), 434–458. doi:10.1037/1082989X.5.4.434

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means and Chapter 6 on trend analysis.)

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

Anderson, S. F., Kelley, K., & Maxwell, S. E. (2017). Sample-size planning for more accurate statistical power: A method adjusting sample effect sizes for publication bias and uncertainty. *Psychological Science, 28*(11), 1547–1562. doi:10.1177/0956797617723724

See Also

ss_aipe_reg_coef, ss_power_R2, ci_nc_F

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(),

ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_s
ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(),
ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(),
ss_power_r(), ss_power_rc(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(),
ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
Cor.Mat <- rbind(
  c(1.00, 0.53, 0.58, 0.60, 0.46, 0.66),
  c(0.53, 1.00, 0.35, 0.07, 0.14, 0.43),
  c(0.58, 0.35, 1.00, 0.18, 0.29, 0.50),
  c(0.60, 0.07, 0.18, 1.00, 0.30, 0.26),
  c(0.46, 0.14, 0.29, 0.30, 1.00, 0.30),
  c(0.66, 0.43, 0.50, 0.26, 0.30, 1.00)
)

rho_XX <- Cor.Mat[2:6, 2:6]
rho_YX <- Cor.Mat[1, 2:6]

# Method 1
ss_power_reg_coef(rho2_Y_X = 0.7826786, rho2_Y_X_without_j = 0.7363697, p = 5,
                  alpha_level = .05, directional = FALSE, desired_power = .80)

# Method 2
ss_power_reg_coef(alpha_level = .05, rho_XX = rho_XX, rho_YX = rho_YX, which_predictor = 5,
                  directional = FALSE, desired_power = .80)

# Method 3
# Here, beta_j is the standardized regression coefficient. Had beta_j
# been the unstandardized regression coefficient, sigma_X and sigma_Y
# would have been the standard deviation for the X variable of
# interest and Y, respectively.
ss_power_reg_coef(rho2_Y_X = 0.7826786, rho2_j_X_without_j = 0.3652136, beta_j = 0.2700964,
                 p = 5, alpha_level = .05, sigma_X = 1, sigma_Y = 1, directional = FALSE,
                  desired_power = .80)

# Method 4
ss_power_reg_coef(alpha_level = .05, cohen_f2 = 0.2130898, p = 5,
                  directional = FALSE, desired_power = .80)

# Power given a specified N and squared multiple correlation coefficients.
ss_power_reg_coef(rho2_Y_X = 0.7826786, rho2_Y_X_without_j = 0.7363697, specified_N = 25,
                  p = 5, alpha_level = .05, directional = FALSE)

# Power given a specified N and effect size.
ss_power_reg_coef(alpha_level = .05, cohen_f2 = 0.2130898, p = 5, specified_N = 25,
                  directional = FALSE)

# Reproducing Maxwell's (2000, p. 445) Example
Cor.Mat.Maxwell <- rbind(
  c(1.00, 0.35, 0.20, 0.20, 0.20, 0.20),
```

```
  c(0.35, 1.00, 0.40, 0.40, 0.40, 0.40),
  c(0.20, 0.40, 1.00, 0.45, 0.45, 0.45),
  c(0.20, 0.40, 0.45, 1.00, 0.45, 0.45),
  c(0.20, 0.40, 0.45, 0.45, 1.00, 0.45),
  c(0.20, 0.40, 0.45, 0.45, 0.45, 1.00)
)

RHO.XX.Maxwell <- Cor.Mat.Maxwell[2:6, 2:6]
Rho.YX.Maxwell <- Cor.Mat.Maxwell[1, 2:6]
R2.Maxwell <- Rho.YX.Maxwell %*% solve(RHO.XX.Maxwell) %*% Rho.YX.Maxwell

RHO.XX.Maxwell.no.1 <- Cor.Mat.Maxwell[3:6, 3:6]
Rho.YX.Maxwell.no.1 <- Cor.Mat.Maxwell[1, 3:6]
R2.Maxwell.no.1 <-
  Rho.YX.Maxwell.no.1 %*% solve(RHO.XX.Maxwell.no.1) %*% Rho.YX.Maxwell.no.1

# This procedure arrives at N = 111, whereas Maxwell (2000, p. 445)
# reports N = 113. The two differ because of the noncentrality
# parameterization, not rounding: this function uses the fixed-predictor
# noncentrality sqrt(N) * f (see Details), while the tabled value rests on
# Cohen's (1988) convention. Neither is a random-predictor result; under
# random predictors, where the predictor values vary across replications,
# the sample size needed for the same power is larger still.
ss_power_reg_coef(rho2_Y_X = R2.Maxwell, rho2_Y_X_without_j = R2.Maxwell.no.1, p = 5,
                  alpha_level = .05, directional = FALSE, desired_power = .80)
```

`ss_power_reg_coef_sensitivity`

Sensitivity Analysis for the Power of a Targeted Regression Coefficient

---

Description

Monte Carlo sensitivity analysis for the statistical power of the *t*-test of a targeted regression coefficient. Given a planned (`estimated_*`) covariance structure and a true (`true_*`) covariance structure, the function draws `G` replications, fits the multiple regression, and reports the empirical proportion of replications on which the *t*-test of the targeted coefficient rejects, together with the realized distribution of $\hat{b}_j$, its standard error, and the test statistic. `ss_power_reg_coef_sensitivity()` is the power-oriented sibling of `ss_aipe_reg_coef_sensitivity` (which is CI-width oriented).

Usage

```
ss_power_reg_coef_sensitivity(
  true_var_Y = NULL,
  true_cov_YX = NULL,
  true_cov_XX = NULL,
  estimated_var_Y = NULL,
  estimated_cov_YX = NULL,
  estimated_cov_XX = NULL,
```

```
  specified_N = NULL,
  which_predictor = 1,
  desired_power = 0.85,
  alpha_level = 0.05,
  directional = FALSE,
  standardize = FALSE,
  G = 1000,
  print_iter = TRUE,
  filename = NULL
)
```

Arguments

`true_var_Y`	Population variance of the dependent variable (Y)
`true_cov_YX`	Population covariance vector between the `p` predictor variables and the dependent variable (Y)
`true_cov_XX`	Population covariance matrix of the `p` predictor variables
`estimated_var_Y`	Estimated variance of the dependent variable (Y) used in sample size planning. Defaults to `true_var_Y`.
`estimated_cov_YX`	Estimated covariance vector between the predictor variables and the dependent variable used in sample size planning. Defaults to `true_cov_YX`.
`estimated_cov_XX`	Estimated covariance matrix of the predictor variables used in sample size planning. Defaults to `true_cov_XX`.
`specified_N`	Directly specified sample size; if supplied, sample size planning is skipped.
`which_predictor`	Index identifying which of the `p` predictors is the targeted predictor for the power test.
`desired_power`	Desired degree of statistical power used for planning
`alpha_level`	Type I error rate
`directional`	Whether a one-sided or two-sided test is used
`standardize`	Whether each replication's data should be standardized prior to fitting (giving a standardized regression coefficient)
`G`	Number of Monte Carlo replications
`print_iter`	Whether to print the iteration number during the simulation
`filename`	Optional path of a CSV file to receive the per-replication results (the coefficient estimate, its standard error, the t statistic, and the observed R^2), overwriting any file already at that path; the default `NULL` writes nothing, and a throwaway run that wants the file should point it at `tempfile(fileext = ".csv")`.

Details

When the estimated and true covariance structures are identical, the function performs a Monte Carlo evaluation of the planning procedure (no misspecification); when they differ, it performs a sensitivity analysis on the consequences of misspecifying the population covariance structure for the targeted coefficient's power. The planning step calls `ss_power_reg_coef` with the estimated covariance structure; the simulation step generates data from the true covariance structure.

Value

A `data.frame` with columns `term` and `value` summarizing the Monte Carlo sensitivity analysis. The `term` entries are: `total_N` (the sample size evaluated), `empirical_power`, `analytic_power` (computed from `ss_power_reg_coef`), the mean / median / SD of the realized $\hat{b}_j$ (`mean_b_j`, `median_b_j`, `sd_b_j`), of its standard error (`mean_se_b_j`, `median_se_b_j`, `sd_se_b_j`), of the test statistic (`mean_t`, `median_t`, `sd_t`), and of the squared multiple correlation coefficient (`mean_R2`, `median_R2`, `sd_R2`), `t_crit` (the critical value), and the input echoes `p`, `which_predictor`, `true_b_j` and `estimated_b_j` (the population and planning values of the targeted coefficient implied by the supplied covariance structures), `desired_power` (NA when `specified_N` was supplied instead), and `alpha_level`. The result carries the `dmar_ss_power_sensitivity` class, so `tidy` reports the planned sample size beside the empirical and analytic power, and `glance` adds the simulated estimator distribution beside the echoed inputs.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Kelley, K., & Maxwell, S. E. (2008). Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects. In P. Alasuutari, L. Bickman, & J. Brannen (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Sage.

Maxwell, S. E. (2000). Sample size and multiple regression analysis. *Psychological Methods, 5*(4), 434–458. doi:10.1037/1082989X.5.4.434

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons of means and Chapter 6 on trend analysis.)

See Also

`ss_power_reg_coef`, `ss_aipe_reg_coef_sensitivity`

`design_consequences` for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_sc, ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# Targeted coefficient power sensitivity with two predictors. The
# default G = 1000 replications is used in practice; G is reduced here
# so the example runs quickly.
set.seed(113)
Sigma_X <- matrix(c(1, 0.3, 0.3, 1), nrow = 2)
cov_YX  <- c(0.4, 0.3)
ss_power_reg_coef_sensitivity(
  true_var_Y = 1, true_cov_YX = cov_YX, true_cov_XX = Sigma_X,
  which_predictor = 1, desired_power = 0.80,
  G = 100, print_iter = FALSE
)
```

---

ss_power_rm_anova *Sample Size or Power for a One-Way Repeated Measures ANOVA Omnibus* F *Test*

Description

Determine the necessary number of subjects to achieve a desired level of statistical power for the omnibus *F* test of the within-subjects factor in a one-way repeated measures ANOVA, or, given a number of subjects, return the realized statistical power.

Usage

```
ss_power_rm_anova(
  a,
  f = NULL,
  eta_squared = NULL,
  rho = 0,
  epsilon = 1,
  desired_power = 0.85,
  alpha_level = 0.05,
  n = NULL
)
```

Arguments

a	Number of measurement occasions (levels of the within-subjects factor)
f	Cohen's f effect size for the within-subjects factor (the population value); supply this or eta_squared, but not both
eta_squared	Population eta squared (proportion of variance, on the relevant scale, accounted for by the within-subjects factor); supply this or f
rho	Average correlation among the repeated measures (default 0). With rho > 0, the within-subjects test gains efficiency relative to a between-subjects analogue
epsilon	Greenhouse-Geisser / Huynh-Feldt sphericity adjustment in (0, 1] (default 1, sphericity assumed). When epsilon < 1, both numerator and denominator degrees of freedom are multiplied by epsilon
desired_power	Desired statistical power (default 0.85)
alpha_level	Type I error rate (default 0.05)
n	Number of subjects (each measured at all a occasions); if specified, returns the realized power

Details

Under the alternative hypothesis with sphericity (epsilon = 1), the within-subjects F statistic follows a noncentral F distribution with numerator df $a - 1$, denominator df $(n - 1)(a - 1)$, and noncentrality parameter $\lambda = naf^2/(1 - \rho)$, where f is Cohen's f for the within-subjects effect and ρ is the average correlation across the repeated measures (Maxwell, Delaney, & Kelley, 2027). Setting rho = 0 reduces to the between-subjects expression.

When sphericity is violated, supplying epsilon (e.g., a Greenhouse-Geisser estimate) rescales the test using the Muller-Barton convention: both numerator and denominator degrees of freedom are multiplied by epsilon, and the noncentrality parameter is likewise multiplied by epsilon. Smaller epsilon therefore reduces power and increases the necessary sample size.

Value

A data.frame with rows for necessary_n_subjects (or specified_n_subjects), a, effect_df, error_df, noncentrality, and actual_power. The result carries the dmar_ss_power class, so tidy and glance summarize it in broom convention (the reported size is the number of subjects).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

See Also

ss_power_one_way_anova, ss_power_pcm

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_s ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_sc(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# 4 measurement occasions, f = 0.25, average within-subject correlation 0.5, power = .80
ss_power_rm_anova(a = 4, f = 0.25, rho = 0.5, desired_power = 0.80)

# Same but with Greenhouse-Geisser epsilon = 0.75
ss_power_rm_anova(a = 4, f = 0.25, rho = 0.5, epsilon = 0.75, desired_power = 0.80)

# Realized power at n = 20 subjects, a = 4
ss_power_rm_anova(a = 4, f = 0.25, rho = 0.5, n = 20)
```

---

ss_power_sc *Sample Size or Power for a Standardized Contrast in a One-Way Between-Subjects ANOVA*

Description

Determine the necessary per-group sample size to achieve a desired level of statistical power for the test of a single planned standardized contrast in a one-way between-subjects analysis of variance, or, given a per-group sample size, return the realized statistical power.

Usage

```
ss_power_sc(
  psi_standardized,
  c_weights,
  desired_power = 0.85,
  alpha_level = 0.05,
  n = NULL,
  directional = FALSE
)
```

Arguments

psi_standardized	The population standardized contrast effect, ψ/σ, where σ is the within-group standard deviation
c_weights	Vector of contrast weights (must sum to zero); use fractional weights so that the positive weights sum to 1 (e.g., c(0.5, 0.5, -0.5, -0.5))
desired_power	Desired statistical power (default 0.85)
alpha_level	Type I error rate (default 0.05)
n	Per-group sample size (assumed balanced); if specified, returns the realized power
directional	Logical: TRUE for a one-sided test (in the same sign as psi_standardized), FALSE (default) for a two-sided test

Details

Under the alternative hypothesis the contrast t-statistic follows a noncentral t-distribution with degrees of freedom $N-J$ (J =length(c_weights)) and noncentrality parameter $\lambda = \psi^*/\sqrt{\sum c_j^2/n}$, where ψ^* is the standardized contrast.

The function searches over per-group sample sizes n until power first reaches desired_power; when n is supplied it returns the realized power.

Value

A data.frame with rows for necessary_n_per_group (or specified_n_per_group), actual_power, and noncentral_t_parm. The result carries the dmar_ss_power class, so tidy and glance summarize it in broom convention.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Lai, K., & Kelley, K. (2012). Accuracy in parameter estimation for ANCOVA and ANOVA contrasts: Sample size planning via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 350–370. doi:10.1111/j.20448317.2011.02029.x

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

ss_power_c, ss_power_one_way_anova, ci_sc, ss_aipe_sc

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(),

ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_se ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sem(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# Power for a standardized contrast of 0.5 across 4 groups,
# contrast (G1 + G2)/2 vs (G3 + G4)/2, desired power = .80
ss_power_sc(psi_standardized = 0.5, c_weights = c(0.5, 0.5, -0.5, -0.5),
            desired_power = 0.80)

# Realized power at n = 30 per group
ss_power_sc(psi_standardized = 0.5, c_weights = c(0.5, 0.5, -0.5, -0.5), n = 30)
```

---

ss_power_sem — *Sample Size Planning for Structural Equation Modeling From the Power Analysis Perspective*

Description

Calculate the necessary sample size for an SEM study, so as to have enough power to reject the null hypothesis that (a) the model has perfect fit, or (b) the difference in fit between two nested models equal some specified amount.

Usage

```
ss_power_sem(
  F_ML = NULL,
  df = NULL,
  RMSEA_null = NULL,
  RMSEA_true = NULL,
  F_full = NULL,
  F_res = NULL,
  RMSEA_full = NULL,
  RMSEA_res = NULL,
  df_full = NULL,
  df_res = NULL,
  alpha_level = 0.05,
  desired_power = 0.85
)
```

Arguments

F_ML	The true maximum likelihood fit function value in the population for the model of interest. Leave this argument NULL if you are doing nested model significance tests
df	The degrees of freedom of the model of interest. Leave this argument NULL if you are doing nested model significance tests
RMSEA_null	The model's population RMSEA under the null hypothesis. Leave this argument NULL if you are doing nested model significance tests
RMSEA_true	The model's population RMSEA under the alternative hypothesis. This should be the model's true population RMSEA value. Leave this argument NULL if you are doing nested model significance tests
F_full	The maximum likelihood fit function value for the full model
F_res	The maximum likelihood fit function value for the restricted model
RMSEA_full	The population RMSEA value for the full model
RMSEA_res	The population RMSEA value for the restricted model
df_full	The degrees of freedom for the full model
df_res	The degrees of freedom for the restricted model
alpha_level	The Type I error rate. Defaults to 0.05.
desired_power	The desired power. Defaults to 0.85, matching the rest of the ss_power_* family.

Value

A data.frame with a necessary_N row, the smallest integer N whose power reaches desired_power under the supplied fit-function or RMSEA alternative, and an actual_power row giving the realized power at that N.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

MacCallum, R. C., Browne, M. W., & Sugawara, H. M. (1996). Power analysis and determination of sample size for covariance structure modeling. *Psychological Methods, 1*(2), 130–149. doi:10.1037/1082989X.1.2.130

Lai, K., & Kelley, K. (2011). Accuracy in parameter estimation for targeted effects in structural equation modeling: Sample size planning for narrow confidence intervals. *Psychological Methods, 16*(2), 127–148. doi:10.1037/a0021764

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

See Also

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_sc ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_smd(), ss_power_split_plot_anova()

Examples

```
# One-model test: necessary N to reject H0: RMSEA = 0 in favor of
# a model whose population RMSEA is 0.05 at 80% power, alpha = .05,
# with df = 20.
ss_power_sem(RMSEA_null = 0, RMSEA_true = 0.05, df = 20,
             alpha_level = 0.05, desired_power = 0.80)

# Equivalent input via the population fit function: F_ML = df * RMSEA^2.
ss_power_sem(F_ML = 20 * 0.05^2, df = 20, alpha_level = 0.05, desired_power = 0.80)

# Two-model nested test: necessary N to detect the difference
# between a full model (RMSEA = 0.04, df = 18) and a restricted
# model (RMSEA = 0.06, df = 22) at 80% power.
ss_power_sem(RMSEA_full = 0.04, df_full = 18,
             RMSEA_res = 0.06, df_res = 22,
             alpha_level = 0.05, desired_power = 0.80)
```

---

ss_power_smd — *Sample Size or Power for a Standardized Mean Difference (Two Independent Groups)*

Description

Determine the necessary per-group sample size to achieve a desired level of statistical power for the two-sample (independent groups) *t*-test on a standardized mean difference (Cohen's *d*; equivalently Hedges' *g* and Glass's *g* for sample size purposes). Alternatively, given a per-group sample size, return the realized statistical power.

Usage

```
ss_power_smd(
  smd,
  desired_power = 0.85,
  alpha_level = 0.05,
```

```
  n_1 = NULL,
  n_2 = NULL,
  directional = FALSE
)
```

Arguments

smd	Supposed standardized mean difference (Cohen's d) the design is planned against: a value the researcher posits for the population, either a minimally important effect or a value believed to be true in the population, never a sample estimate. Echoed in the returned table as the supposed_smd row.
desired_power	Desired statistical power (default 0.85)
alpha_level	Type I error rate (default 0.05)
n_1	Sample size for group 1 (if specified, the function returns the realized power; assumes n_2 = n_1 unless n_2 is also given)
n_2	Sample size for group 2 (defaults to n_1 when n_1 is supplied)
directional	Logical: TRUE for a one-sided test (in the same sign as smd), FALSE (default) for a two-sided test

Details

The two-sample t-statistic with pooled standard deviation follows a noncentral t-distribution with $n_1 + n_2 - 2$ degrees of freedom and noncentrality parameter $\lambda = \delta\sqrt{n_1 n_2/(n_1 + n_2)}$, where δ is the population standardized mean difference. For balanced designs ($n_1 = n_2 = n$) this simplifies to $\lambda = \delta\sqrt{n/2}$.

Power is computed as the probability that the absolute value of the test statistic exceeds the critical value(s) under the alternative; the function returns the per-group sample size for which power first reaches desired_power.

Kelley and Rausch (2006) develop the accuracy in parameter estimation approach to planning the sample size for the standardized mean difference, implemented in ss_aipe_smd.

Value

A data.frame with term and value columns. The design result comes first, followed by rows that echo the user-supplied planning inputs, so the assumptions the power was evaluated under travel with the result. The supposed_smd row is the supposed effect the plan is built on: a value the researcher posits, either a minimally important effect or a value believed to be true in the population, never a sample estimate. The tails row is 2 for a nondirectional test and 1 for a directional test.

When n_1 **is** NULL Result rows necessary_n_per_group, actual_power, and noncentral_t_parm, then the planning inputs supposed_smd, desired_power, alpha_level, and tails.

When n_1 **is specified** Result rows specified_n_1, specified_n_2, actual_power, and noncentral_t_parm, then the planning inputs supposed_smd, alpha_level, and tails (the supplied group sizes are the specified_n_1 / specified_n_2 rows).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Kelley, K., Maxwell, S. E., & Rausch, J. R. (2003). Obtaining power or obtaining precision: Delineating methods of sample size planning. *Evaluation and the Health Professions, 26*(3), 258–287. doi:10.1177/0163278703255242

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Maxwell, S. E., Kelley, K., & Rausch, J. R. (2008). Sample size planning for statistical power and accuracy in parameter estimation. *Annual Review of Psychology, 59*, 537–563. doi:10.1146/annurev.psych.59.103006.093735

See Also

ss_aipe_smd, ci_smd, smd, ci_nc_t

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_s, ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_split_plot_anova()

Examples

```
# Per-group sample size for d = 0.5, alpha = .05, power = .80, two-sided
ss_power_smd(smd = 0.5, desired_power = 0.80)

# Same with a directional (one-sided) test
ss_power_smd(smd = 0.5, desired_power = 0.80, directional = TRUE)

# Realized power given balanced n = 30 per group
ss_power_smd(smd = 0.5, n_1 = 30)

# Realized power for unbalanced (n_1 = 30, n_2 = 50)
ss_power_smd(smd = 0.5, n_1 = 30, n_2 = 50)
```

`ss_power_split_plot_anova` *Sample Size or Power for a Mixed-Effects ANOVA (Between X Within Design)*

Description

Determine the necessary per-group sample size to achieve a desired level of statistical power for one of the three *F* tests in a mixed-effects ANOVA with one between-subjects factor and one within-subjects factor – between-subjects main effect, within-subjects main effect, or the between x within interaction – or, given a per-group sample size, return the realized statistical power. (This design is also commonly called a split-plot factorial.)

Usage

```
ss_power_split_plot_anova(
  a,
  b,
  effect,
  f = NULL,
  partial_eta_squared = NULL,
  rho,
  epsilon = 1,
  desired_power = 0.85,
  alpha_level = 0.05,
  n = NULL
)
```

Arguments

`a`	Number of levels of the between-subjects factor (i.e., number of groups)
`b`	Number of levels of the within-subjects factor (i.e., number of measurement occasions)
`effect`	Which *F* test to compute power for: `"between"`, `"within"`, or `"interaction"`
`f`	Cohen's *f* effect size for the chosen effect (the population value); supply this or `partial_eta_squared`, but not both
`partial_eta_squared`	Partial eta squared for the chosen effect; supply this or `f`
`rho`	Average correlation among the repeated measures within a subject (must lie in (-1, 1)). Higher `rho` reduces power for the between-subjects test (because subject means contain more redundant information) and increases power for the within-subjects and interaction tests
`epsilon`	Greenhouse-Geisser / Huynh-Feldt sphericity adjustment in (0, 1] (default 1, sphericity assumed). Applied to the within-subjects and interaction tests but not the between-subjects test. Both numerator and denominator df, and the noncentrality, are multiplied by `epsilon` (Muller-Barton convention)

desired_power	Desired statistical power (default 0.85)
alpha_level	Type I error rate (default 0.05)
n	Per-group (between-subjects) sample size; if specified, returns the realized power

Details

This is a two-factor mixed-effects design: one between-subjects factor with a levels and one within-subjects factor with b levels; n subjects are randomly assigned to each between-subjects level and each subject is measured at all b within-subjects levels, for $N = na$ subjects total and Nb observations. The covariance among the b within-subject observations is summarized by rho, the average pairwise correlation.

The three *F* tests have noncentrality parameters

$$\lambda_B = Nbf^2/(1 + (b-1)\rho)$$

for the between-subjects test (numerator df $a - 1$, denominator df $N - a$),

$$\lambda_W = Nbf^2\,\epsilon/(1-\rho)$$

for the within-subjects test (numerator df $(b-1)\epsilon$, denominator df $(N-a)(b-1)\epsilon$), and the same form as λ_W for the interaction (numerator df $(a-1)(b-1)\epsilon$, same denominator df). Cohen's f relates to partial eta squared via $f = \sqrt{\eta_p^2/(1-\eta_p^2)}$.

This design is the compound-symmetry (random intercept) special case of the two-level linear mixed-effects model: rho is the intraclass correlation and the b occasions are the level-1 units of a subject. For two between-subjects groups the between-subjects $F(1, .)$ test is therefore the two-level treatment *t* test of ss_power_mixed_effects squared, so the two planners agree on that shared case.

Value

A data.frame with rows for necessary_n_per_group (or specified_n_per_group), total_N, effect_df, error_df, noncentrality, and actual_power. The result carries the dmar_ss_power class, so tidy and glance summarize it in broom convention (the reported size is the per-group count).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Muller, K. E., & Barton, C. N. (1989). Approximate power for repeated measures ANOVA lacking sphericity. *Journal of the American Statistical Association, 84*, 549–555.

See Also

ss_power_one_way_anova, ss_power_factorial_anova, ss_power_rm_anova, ss_power_mixed_effects

design_consequences for what a chosen design delivers: power, the Type S (sign) and Type M (exaggeration) errors of the significance filter, and the expected confidence interval width.

Other sample size for power: power_fisher_exact(), ss_aipe_mixed_effects(), ss_power_R2(), ss_power_R2_sensitivity(), ss_power_c(), ss_power_c_ancova(), ss_power_composite_ancova(), ss_power_composite_ancova_2group(), ss_power_composite_anova(), ss_power_composite_factorial_ancova(), ss_power_composite_factorial_ancova_het(), ss_power_composite_factorial_anova(), ss_power_composite_s ss_power_contrast(), ss_power_equivalence_c(), ss_power_factorial_ancova(), ss_power_factorial_anova(), ss_power_indirect_effect(), ss_power_mixed_effects(), ss_power_one_way_anova(), ss_power_pcm(), ss_power_r(), ss_power_rc(), ss_power_reg_coef(), ss_power_reg_coef_sensitivity(), ss_power_rm_anova(), ss_power_sc(), ss_power_sem(), ss_power_smd()

Other mixed models: R2_mixed_effects(), R2_mixed_effects_decomposition(), icc_lmer(), manova_split_plot(), mixed_anova(), ss_aipe_mixed_effects(), ss_aipe_mixed_effects_sensitivity(), ss_power_mixed_effects()

Examples

```
# 2 groups, 4 occasions, between-subjects effect, f = 0.25,
# average within-subject correlation 0.5, power = .80
ss_power_split_plot_anova(a = 2, b = 4, effect = "between", f = 0.25,
                    rho = 0.5, desired_power = 0.80)

# Same design, within-subjects (occasion) main effect, f = 0.25
ss_power_split_plot_anova(a = 2, b = 4, effect = "within", f = 0.25,
                    rho = 0.5, desired_power = 0.80)

# Same design, between x within interaction, f = 0.25
ss_power_split_plot_anova(a = 2, b = 4, effect = "interaction", f = 0.25,
                    rho = 0.5, desired_power = 0.80)

# Realized power for n = 25 per group on the interaction test, partial eta^2 = 0.06
ss_power_split_plot_anova(a = 2, b = 4, effect = "interaction",
                    partial_eta_squared = 0.06, rho = 0.5, n = 25)

# Greenhouse-Geisser correction with epsilon = 0.7 on the within-subjects test
ss_power_split_plot_anova(a = 2, b = 4, effect = "within", f = 0.25,
                    rho = 0.5, epsilon = 0.7, desired_power = 0.80)
```

---

ss_seq_c *Sequential Sample Size for a Fixed-Width Contrast Interval*

Description

Implements the purely sequential fixed-width confidence interval procedure for a linear contrast $\psi = \sum_j c_j \mu_j$, following Chattopadhyay, Bandyopadhyay, Kelley, and Padalunkal (2025). The goal

is a 100(1 - 2α)% confidence interval $\hat{\psi} \pm h$ whose half-width h is fixed in advance, which is what makes a noninferiority or equivalence verdict reachable by design: with bounds $\pm\delta$, equivalence can never be declared unless $h < \delta$, and targeting $h = \delta/2$ gives a truly equivalent contrast about a 90% chance of being declared equivalent at $\alpha = .05$. Because no fixed sample size can guarantee a bounded-width interval when the error variance is unknown (Dantzig, 1940), the procedure is sequential: begin with a pilot, then keep sampling, re-estimating the variance, until the stopping criterion is met. Used with `pilot = TRUE` the function plans the pilot and the cost-optimal allocation; with `pilot = FALSE` it evaluates the stopping criterion at the data in hand, in the style of `mr_smd`.

Usage

```
ss_seq_c(
  c_weights,
  half_width,
  s = NULL,
  n = NULL,
  cost = NULL,
  alpha_level = 0.05,
  quantile = c("t", "normal"),
  pilot = FALSE,
  m0 = 10
)
```

Arguments

`c_weights`	The contrast weights. The weights must sum to zero with the positive weights summing to 1 and the negative weights to -1, so that `half_width` is on the raw scale of the response.
`half_width`	The target half-width h of the 100(1 - 2α)% confidence interval, in raw units of the response. For a noninferiority or equivalence decision at bounds $\pm\delta$, the recommended target is $\delta/2$.
`s`	The current estimate(s) of the error standard deviation: either a single pooled value or one value per group (aligned with `c_weights`). Required when `pilot = FALSE`; optional planning values when `pilot = TRUE` (used only to shape the allocation).
`n`	The current per-group sample sizes, aligned with `c_weights`. Required when `pilot = FALSE`.
`cost`	Optional per-observation sampling costs, one per group, aligned with `c_weights`. When supplied, the reported allocation is the cost-optimal $n_j \propto \lvert c_j \rvert \, \sigma_j / \sqrt{\text{cost}_j}$ (Chattopadhyay et al., 2025); with equal costs and standard deviations this reduces to allocation proportional to $\lvert c_j \rvert$.
`alpha_level`	One-sided rate per bound; the interval is at confidence level 1 - 2α. Default `0.05` (a 90% interval).
`quantile`	`"t"` (default) uses the *t* quantile on the current error degrees of freedom in the stopping criterion, a finite-sample refinement; `"normal"` uses the normal quantile of the classical statement of the rule (Chow & Robbins, 1965). The normal rule stops slightly early at wide targets, the finite-sample undershoot anticipated by Woodroofe (1977); the *t* rule corrects it at a small cost in sample size.

`pilot`	`TRUE` to plan the pilot stage; `FALSE` (default) to evaluate the stopping criterion at the current data.
`m0`	The minimum pilot sample size per group. Default `10`. A pilot that is too small makes the variance estimate that drives the early steps unstable.

Details

The stopping criterion. Sampling stops at the first N for which

$$q^2 \sum_j c_j^2 s_j^2 / n_j \;\le\; h^2,$$

where q is the t or normal quantile at $1 - \alpha$. With a pooled s and equal allocation this is the Chow and Robbins (1965) rule specialized to a contrast; the procedure is asymptotically consistent (coverage approaches 1 - 2α) and first-order efficient (the mean stopping size approaches the oracle n^* an investigator with known variance would use). See `ss_seq_c_sensitivity` for a Monte Carlo evaluation of both properties.

Degrees of freedom. With a pooled `s` the criterion uses $\nu = N - J$. With per-group `s` it uses the Satterthwaite approximation, which is the appropriate error law when the variances are not assumed homogeneous.

Batches. Observations may be added in batches rather than one at a time; the asymptotic properties survive batching. Re-call the function after each batch.

Value

With `pilot = TRUE`, a `data.frame` with row `pilot_n_per_group` (the pilot size for each group with a nonzero weight) followed by one `allocation_j` row per group giving the recommended sampling proportions for the accrual stage. With `pilot = FALSE`, rows `stop` (1 = the criterion is met, stop sampling; 0 = continue), `half_width_current` (the half-width the interval would have now), `half_width_target`, `N_current`, `N_projected` (the approximate total at which the criterion would be met under the current allocation, from the normal approximation), and the `allocation_j` rows for the next round of sampling.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Chattopadhyay, B., Bandyopadhyay, T., Kelley, K., & Padalunkal, J. J. (2025). A sequential approach for noninferiority or equivalence of a linear contrast under cost constraints. *Psychological Methods, 30*(2), 425–439. doi:10.1037/met0000570

Chow, Y. S., & Robbins, H. (1965). On the asymptotic theory of fixed-width sequential confidence intervals for the mean. *The Annals of Mathematical Statistics, 36*(2), 457–462.

Dantzig, G. B. (1940). On the non-existence of tests of "Student's" hypothesis having power functions independent of σ. *The Annals of Mathematical Statistics, 11*(2), 186–192.

Mukhopadhyay, N., & de Silva, B. M. (2009). *Sequential methods and their applications*. CRC Press.

Woodroofe, M. (1977). Second order approximations for sequential point and interval estimation. *The Annals of Statistics, 5*(5), 984–995.

See Also

ss_seq_c_sensitivity, ss_aipe_c, ss_power_equivalence_c, equivalence_c, mr_smd

Other sequential estimation: ss_seq_c_sensitivity()

Examples

```
# 1. Plan the pilot for a two-group contrast, target half-width 2.5
#    (bounds of 5 with the h = delta/2 rule):
ss_seq_c(c_weights = c(1, -1), half_width = 2.5, pilot = TRUE)

# 2. Evaluate the stopping criterion mid-study: pooled s = 15.4 at
#    n = 60 per group. Too imprecise to stop; the projection says
#    roughly how much further to go.
ss_seq_c(c_weights = c(1, -1), half_width = 2.5,
         s = 15.4, n = c(60, 60))

# 3. Costs differ: sampling the second group costs four times as
#    much per observation, so its share of new observations drops.
ss_seq_c(c_weights = c(1, -1), half_width = 2.5,
         s = c(15.4, 15.4), n = c(60, 60), cost = c(1, 4))
```

---

ss_seq_c_sensitivity *Monte Carlo Sensitivity of the Sequential Fixed-Width Procedure*

Description

Simulates the purely sequential fixed-width confidence interval procedure of ss_seq_c under a known data generating mechanism, reporting the distribution of the stopping sample size and the empirical coverage of the fixed-width interval. The two quantities to read are the ratio of the mean stopping size to the oracle n^* (first-order efficiency: the ratio approaches 1 as the target half-width shrinks) and the coverage (asymptotic consistency: coverage approaches 1 - 2α). The normal quantile rule stops slightly early at wide targets, the finite-sample undershoot anticipated by Woodroofe (1977); the *t* quantile rule corrects it at a small cost in sample size.

Usage

```
ss_seq_c_sensitivity(
  c_weights,
  half_width,
  true_sigma,
  true_means = NULL,
  alpha_level = 0.05,
  quantile = c("t", "normal"),
```

```
  m0 = 10,
  G = 1000,
  seed = NULL
)
```

Arguments

c_weights	The contrast weights. The weights must sum to zero with the positive weights summing to 1 and the negative weights to -1.
half_width	The target half-width h of the 100(1 - 2α)% interval, in raw units of the response.
true_sigma	The data generating error standard deviation: a single value applied to every group.
true_means	Optional vector of data generating group means, aligned with c_weights. Default all zero, so the true contrast is 0.
alpha_level	One-sided rate per bound; the interval is at confidence level 1 - 2α. Default 0.05.
quantile	"t" (default) or "normal"; see ss_seq_c.
m0	Pilot sample size per group. Default 10.
G	Number of Monte Carlo replications. Default 1000.
seed	Optional integer seed. Default NULL (the current RNG state is used). When supplied, the caller's RNG state is restored on exit.

Details

Simulation design. Each replication samples the groups with nonzero weights in balanced fashion (one observation per group per step) from normal populations with common true_sigma, starting at m0 per group, and stops at the first step satisfying the ss_seq_c criterion with the pooled variance estimate. This matches the equal-cost, equal-variance case of Chattopadhyay, Bandyopadhyay, Kelley, and Padalunkal (2025); unequal costs change the optimal allocation but not the logic.

The oracle. With known σ and balanced allocation over the J_0 groups with nonzero weights, the fixed-width requirement is $n^* = z^2_{1-\alpha}\,\sigma^2 J_0 \sum_j c_j^2 / h^2$ in total. The sequential procedure spends about n^* without knowing σ, which is its point.

Value

A data.frame with rows n_star (the oracle total sample size an investigator with known σ would use), mean_N, median_N, sd_N (the stopping total across replications), ratio_mean_N_n_star, coverage (the proportion of replications whose $\hat{\psi}_N \pm h$ interval covered the true contrast), se_coverage (its simulation standard error), and the input echoes half_width, true_psi (the population contrast implied by c_weights and true_means), true_sigma, alpha_level, and m0.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Chattopadhyay, B., Bandyopadhyay, T., Kelley, K., & Padalunkal, J. J. (2025). A sequential approach for noninferiority or equivalence of a linear contrast under cost constraints. *Psychological Methods, 30*(2), 425–439. doi:10.1037/met0000570

Chow, Y. S., & Robbins, H. (1965). On the asymptotic theory of fixed-width sequential confidence intervals for the mean. *The Annals of Mathematical Statistics, 36*(2), 457–462.

Ghosh, M., Mukhopadhyay, N., & Sen, P. K. (1997). *Sequential estimation*. Wiley.

Woodroofe, M. (1977). Second order approximations for sequential point and interval estimation. *The Annals of Statistics, 5*(5), 984–995.

See Also

`ss_seq_c`, `ss_aipe_c`, `ss_power_equivalence_c`

Other sequential estimation: `ss_seq_c()`

Examples

```
# A two-group contrast, target half-width 2.5, error SD 15.67:
# the t-quantile rule stops near the oracle with near-nominal
# coverage. (G kept small here for speed; use G = 2000 or more in
# earnest.)
ss_seq_c_sensitivity(c_weights = c(1, -1), half_width = 2.5,
                     true_sigma = 15.67, G = 200, seed = 113)
```

---

summary_t_test *Two-Sample* t *Test From Summary Statistics*

Description

Computes a two-sample *t* test (pooled or Welch) directly from the per-group means, standard deviations, and sample sizes, without requiring access to the raw observations. Returns the test statistic, degrees of freedom, *p*-value, and a CI on the mean difference in a `data.frame`. Useful for re-analyses from published papers that report only the summary numbers.

Usage

```
summary_t_test(
  mean_1,
  sd_1,
  n_1,
  mean_2,
  sd_2,
  n_2,
  mu = 0,
  var_equal = TRUE,
```

```
  alternative = c("two_sided", "less", "greater"),
  conf_level = 0.95
)
```

Arguments

`mean_1`, `mean_2`	Group sample means.
`sd_1`, `sd_2`	Group sample standard deviations.
`n_1`, `n_2`	Group sample sizes.
`mu`	Null value of the mean difference $\mu_1 - \mu_2$. Default `0`.
`var_equal`	Logical. If `TRUE` (default), uses Student's pooled-variance *t*. If `FALSE`, uses Welch's separate- variance *t* with Satterthwaite degrees of freedom.
`alternative`	One of `"two_sided"` (default; the base-R spelling `"two.sided"` is accepted as an alias), `"less"`, or `"greater"`.
`conf_level`	Confidence level for the CI on the mean difference. Default `0.95`.

Details

Pooled-variance *t* (Student, 1908). Under $\sigma_1 = \sigma_2$, the pooled SD is $s_p = \sqrt{((n_1 - 1)s_1^2 + (n_2 - 1)s_2^2)/(n_1 + n_2 - 2)}$, the test statistic is $t = (\bar{x}_1 - \bar{x}_2 - \mu_0)/(s_p\sqrt{1/n_1 + 1/n_2})$, and $df = n_1 + n_2 - 2$.

Welch's *t* (Welch, 1947). Under unequal variances, $t = (\bar{x}_1 - \bar{x}_2 - \mu_0)/\sqrt{s_1^2/n_1 + s_2^2/n_2}$ with Satterthwaite degrees of freedom (see `welch_t`).

Choosing pooled vs Welch. Methodological reviews now recommend Welch as the default (Delacre, Lakens, & Leys, 2017; Ruxton, 2006). Pooled-variance *t* is preserved here primarily for reproducing analyses from older sources that used it.

Value

A `data.frame` with rows for the mean difference, the *t* statistic, degrees of freedom, *p*-value, and the CI lower and upper limits on the mean difference.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Delacre, M., Lakens, D., & Leys, C. (2017). Why psychologists should by default use Welch's *t*-test instead of Student's *t*-test. *International Review of Social Psychology, 30*(1), 92–101. doi:10.5334/irsp.82

Ruxton, G. D. (2006). The unequal variance *t*-test is an underused alternative to Student's *t*-test and the Mann-Whitney *U* test. *Behavioral Ecology, 17*(4), 688–690. doi:10.1093/beheco/ark016

Snedecor, G. W., & Cochran, W. G. (1989). *Statistical methods* (8th ed.). Iowa State University Press.

Student. (1908). The probable error of a mean. *Biometrika, 6*(1), 1–25. doi:10.2307/2331554

Welch, B. L. (1947). The generalization of "Student's" problem when several different population variances are involved. *Biometrika, 34*(1/2), 28–35.

See Also

welch_t, t.test, smd

Other hypothesis tests: adjusted_means(), ancova(), anova_within(), ci_dunnett(), ci_scheffe(), ci_tukey_kramer(), compare_cov_structures(), contrast_test(), correlations_test(), equivalence_r(), equivalence_smd(), factorial_anova(), manova_split_plot(), mauchly_test(), mixed_anova(), obrien_test(), pairwise_within(), randomization_test(), randomization_test_paired(), regions_of_significance(), simple_effects_AB(), welch_t()

Examples

```
# 1. Re-analysis from published summary statistics:
#        Group A: M = 100, SD = 15, n = 30
#        Group B: M = 108, SD = 18, n = 25
summary_t_test(mean_1 = 100, sd_1 = 15, n_1 = 30,
               mean_2 = 108, sd_2 = 18, n_2 = 25)

# 2. Welch version for the same data:
summary_t_test(mean_1 = 100, sd_1 = 15, n_1 = 30,
               mean_2 = 108, sd_2 = 18, n_2 = 25,
               var_equal = FALSE)
```

---

summary_t_test_broom *Broom-Style Tidy / Glance Methods for* summary_t_test()

Description

tidy() returns the single mean-difference estimate and its confidence interval in the **broom** convention; glance() coincides with it, since a two-sample *t* test reports one estimand and there are no extra model-level statistics to add.

Usage

```
## S3 method for class 'dmar_summary_t_test'
tidy(x, ...)

## S3 method for class 'dmar_summary_t_test'
glance(x, ...)
```

Arguments

x	A dmar_summary_t_test object returned by summary_t_test.
...	Unused.

Value

A one-row data.frame with columns term, estimate, ci_lower, ci_upper, statistic, df, p_value, and conf_level.

Author(s)

Ken Kelley <kkelley@nd.edu>

`teacher_expectancy` *Teacher Expectancy Meta-Analysis Data (Raudenbush, 1984)*

Description

The 19 effect sizes from Raudenbush's (1984) synthesis of 18 experiments testing the effect of teacher expectancy on pupil IQ, the meta-analysis that resolved the controversy started by *Pygmalion in the Classroom* (Rosenthal & Jacobson, 1968; the single famous study is shipped separately as `pygmalion`). In each experiment, teachers were told that randomly selected children were likely to bloom intellectually; the synthesis asks how large the resulting IQ advantage was and, centrally, why it varied across studies. Raudenbush's hypothesis, strongly supported, was that the longer teachers had known their pupils before the expectancy induction, the smaller the effect: credible deception is the Achilles' heel of the design.

Usage

```
teacher_expectancy
```

Format

A data frame with 19 rows (18 experiments; Pellegrini and Hicks, 1972, contributes a tester-aware and a tester-blind condition) and 10 variables.

`study` Integer identifier, in the order of the paper's Table 1.

`author` Study authors (with the Pellegrini and Hicks condition noted).

`year` Year of publication.

`weeks` Estimated weeks of teacher-student contact prior to the expectancy induction, 0 to 24. The moderator at the heart of the paper.

`testing` Factor: `group` or `individual` IQ testing.

`tester` Factor: test administrator `aware` of or `blind` to the expectancy designations.

`n_experimental`, `n_control` Per-condition sample sizes (from the studies as tabulated in Raudenbush & Bryk, 1985; the 1984 table does not print them).

`d` Standardized mean difference: the treatment effect in IQ points divided by the control group's posttest standard deviation (positive when the expectancy children gained more). The 1984 paper's Table 1 values.

`p_one_tailed` One-tailed *p*-value reported for the study's expectancy effect.

Details

The study-level Pellegrini and Hicks values. For analyses with the 18 *studies* as units (the combined significance tests, the contrast on weeks of prior contact, and the heterogeneity statistic), Raudenbush merged the two Pellegrini and Hicks conditions into a single study-level entry with $d = 0.52$ and one-tailed $p = .010$ (Table 1 prints these on the study's header row above the two condition rows). Replace rows 4 and 5 with that pair to reconstruct his 18-study analyses, as the teacher expectancy vignette does. For the tester aware-versus-blind comparisons the two conditions enter separately, which is why the data ship at the condition level.

Relation to the 1985 version. Raudenbush and Bryk (1985) re-standardized the same literature for their empirical Bayes analysis (that version circulates as `dat.raudenbush1985` in **metafor**), so its effect sizes differ from the `d` column here, which preserves the 1984 paper's metric. The sample sizes are common to both.

The teacher expectancy vignette (`vignette("teacher_expectancy", package = "DMAR")`) reproduces the paper's analyses with `combine_p`, `meta_contrast`, and `meta_smd`, and then reanalyzes the data with modern random effects machinery.

Author(s)

Ken Kelley <kkelley@nd.edu>

Source

Raudenbush, S. W. (1984). Magnitude of teacher expectancy effects on pupil IQ as a function of the credibility of expectancy induction: A synthesis of findings from 18 experiments. *Journal of Educational Psychology, 76*(1), 85–97.

References

Raudenbush, S. W. (1984). Magnitude of teacher expectancy effects on pupil IQ as a function of the credibility of expectancy induction: A synthesis of findings from 18 experiments. *Journal of Educational Psychology, 76*(1), 85–97. doi:10.1037/00220663.76.1.85

Raudenbush, S. W., & Bryk, A. S. (1985). Empirical Bayes meta-analysis. *Journal of Educational Statistics, 10*(2), 75–98.

Rosenthal, R., & Jacobson, L. (1968). *Pygmalion in the classroom: Teacher expectation and pupils' intellectual development*. Holt, Rinehart and Winston.

See Also

`pygmalion` for the single Rosenthal and Jacobson study this literature grew from; `meta_smd`, `meta_contrast`, and `combine_p` for the analyses the vignette reproduces.

Examples

```
data(teacher_expectancy)
head(teacher_expectancy)

# The paper's central picture: effect size against weeks of prior contact.
plot(d ~ weeks, data = teacher_expectancy,
```

```
     xlab = "Weeks of teacher-student contact before induction",
     ylab = "Effect size d")

# The study-level (18-study) data Raudenbush used for the combined tests:
# merge the two Pellegrini & Hicks conditions into their study row.
study_level <- teacher_expectancy[-c(4, 5), ]
ph <- data.frame(study = 4, author = "Pellegrini & Hicks", year = 1972,
                 weeks = 0, testing = "group", tester = "aware",
                 n_experimental = 22, n_control = 22,
                 d = 0.52, p_one_tailed = .010)
study_level <- rbind(study_level[1:3, ], ph, study_level[4:17, ])
round(c(mean = mean(study_level$d), sd = sd(study_level$d)), 2)  # .11, .20
```

`test_market` *Controlled Test-Market Experiment (Bryant & Bruvold, 1980)*

Description

The controlled test-market experiment of Bryant and Bruvold (1980), used to illustrate multiple-comparison procedures in the analysis of covariance (ANCOVA) when the covariate is *random*. A company compared $k = 6$ marketing strategies ("panels") for a brand, randomly assigning them to retail outlets within $s = 4$ blocks of outlets that were homogeneous in size, locality, and ownership (a randomized complete block design, one outlet per panel-by-block cell). During the experiment a concomitant variable, the remaining category movement in each outlet, becomes available; it cannot be controlled by the experimenter and is best modeled as a random covariate. Adjusting brand movement for this covariate sharply reduces unexplained error, permitting far finer comparison of the panels than the raw outcome allows.

Usage

```
test_market
```

Format

A data frame with 24 observations (6 panels $\times$ 4 blocks) on 4 variables.

`panel` Factor with levels 1–6: the marketing strategy (treatment) randomly assigned to the outlet. Different panels entail different methods of packaging, displaying, or pricing.

`block` Factor with levels 1–4: the block of retail outlets, grouped to be homogeneous in size, locality, ownership, and other considerations that influence brand movement.

`brand_movement` Test-brand movement during the test period, in hundreds of statistical cases. The dependent variable (y in the source).

`category_movement` Remaining category movement, the random concomitant variable (covariate; x in the source). It is not identically distributed across blocks, which is precisely the setting Bryant and Bruvold's grouped-covariate extension was designed for.

Details

Why it is a benchmark for ANCOVA multiple comparisons. The model fitted by Bryant and Bruvold (their Eq. 3.1) is a randomized-block ANCOVA,

$$y_{ij} = \theta_i + \beta_j + (x_{ij} - \delta_j)\, u + e_{ij},$$

with θ_i the ith panel (adjusted) mean, β_j the jth block effect, and u the within-cell covariate slope. The point of the example is that the studentized range of the adjusted panel means does *not* follow the ordinary Tukey distribution, because the covariate is random and its adjustment must be estimated; the correct reference distribution is the Bryant–Paulson generalized studentized range (`bryant_paulson`).

Reproducible quantities. Fitting `lm(brand_movement ~ panel + block + category_movement)` gives a covariate slope of 0.4079 and an error mean square of 0.01326 on $\nu = 14$ degrees of freedom, with adjusted panel means $3.595, 3.619, 4.102, 4.515, 4.618, 4.876$, exactly the values reported in the paper. With $q_{.05;\,1,6,14} = 4.83$ (`qbryant_paulson`), every pairwise simultaneous 95% interval is a difference of adjusted panel means plus or minus 0.278, so two panels differ at the simultaneous 95% level exactly when their adjusted means are more than 0.278 apart. Had the covariate not been measured, the error mean square would have been 0.2368, roughly eighteen times larger, and the intervals about four times wider. See `data-raw/test_market.R` for the construction script and its verification checks.

Author(s)

Ken Kelley

Source

Bryant, J. L., & Bruvold, N. T. (1980). Multiple comparison procedures in the analysis of covariance. *Journal of the American Statistical Association, 75*(372), 874–880 (Table 1). doi:10.2307/2287175

References

Bryant, J. L., & Paulson, A. S. (1976). An extension of Tukey's method of multiple comparisons to experimental designs with random concomitant variables. *Biometrika, 63*, 631–638.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9.)

See Also

`ci_c_ancova_bp` for the simultaneous intervals this data set illustrates, `bryant_paulson` for the critical values, and `ancova` for an ANCOVA fit.

Examples

```
data(test_market)
str(test_market)

# Reproduce the published ANCOVA (slope 0.4079, error MS 0.01326, df 14).
```

```
fit <- lm(brand_movement ~ panel + block + category_movement,
          data = test_market)
coef(fit)["category_movement"]
sum(residuals(fit)^2) / fit$df.residual

# Adjusted panel means at the covariate grand mean.
xbar <- mean(test_market$category_movement)
adj <- vapply(levels(test_market$panel), function(p) {
  nd <- data.frame(panel = factor(p, levels = levels(test_market$panel)),
                   block = factor(1:4, levels = levels(test_market$block)),
                   category_movement = xbar)
  mean(predict(fit, nd))
}, numeric(1))
adj  # 3.595 3.619 4.102 4.515 4.618 4.876

# Bryant-Paulson simultaneous 95% intervals (s = 4 blocks => n = 4,
# df 14), every one of them the difference plus or minus 0.278.
ci_c_ancova_bp(adj_means = adj, s_ancova = sqrt(0.01326),
               n = 4, num_covariates = 1, df = 14)
```

`tidy.dmar_cfa_k` *Tidy a Multiple-Factor CFA Fit*

Description

Returns the parameter rows of a `cfa_k` table (loadings, error variances, intercepts, latent correlations, and the defined measurement properties) in the column convention used by the **broom** ecosystem. Fit-information rows belong in `glance.dmar_cfa_k`.

Usage

```
## S3 method for class 'dmar_cfa_k'
tidy(x, ...)
```

Arguments

`x`	A `dmar_cfa_k` object returned by `cfa_k`.
`...`	Unused.

Value

A `data.frame` with columns `term`, `estimate`, `se`, `statistic`, `p_value`, `ci_lower`, `ci_upper`.

Author(s)

Ken Kelley <kkelley@nd.edu>

Examples

```
data(holzinger_swineford)
res <- cfa_k(holzinger_swineford,
             list(verbal = c("t6_paragraph_comprehension",
                             "t7_sentence", "t9_word_meaning"),
                  deduction = c("t20_deduction",
                                "t22_problem_reasoning",
                                "t23_series_completion")))
generics::tidy(res)
generics::glance(res)
```

`tidy.dmar_ci_R2` *Tidy / Glance Methods for ci_R2 Output*

Description

Returns the standard broom-style one-row summary of the R^2 confidence interval: `term` (always `"R2"`), `estimate`, `ci_lower`, `ci_upper`, `conf_level`.

Usage

```
## S3 method for class 'dmar_ci_R2'
tidy(x, ...)

## S3 method for class 'dmar_ci_R2'
glance(x, ...)
```

Arguments

`x`	A `dmar_ci_R2` object returned by `ci_R2`.
`...`	Unused.

Value

A one-row `data.frame` in broom convention.

Author(s)

Ken Kelley <kkelley@nd.edu>

Examples

```
res <- ci_R2(R2 = 0.25, N = 100, p = 5)
generics::tidy(res)
generics::glance(res)
```

`tidy.dmar_ci_smd` *Tidy / Glance Methods for ci_smd Output*

Description

Returns the standard broom-style one-row summary of the SMD confidence interval: `term` (always `"smd"`), `estimate`, `ci_lower`, `ci_upper`, `conf_level`.

Usage

```
## S3 method for class 'dmar_ci_smd'
tidy(x, ...)

## S3 method for class 'dmar_ci_smd'
glance(x, ...)
```

Arguments

`x`	A `dmar_ci_smd` object returned by `ci_smd`.
`...`	Unused.

Value

A one-row `data.frame` with columns `term` (always `"smd"`), `estimate` (the point estimate of d), `ci_lower` (the lower confidence limit on δ), `ci_upper` (the upper confidence limit on δ), and `conf_level` (the confidence level used in construction; `NA` when asymmetric `alpha_lower` / `alpha_upper` were supplied instead). Column names follow the broom convention so the result composes with the broom ecosystem.

Author(s)

Ken Kelley <kkelley@nd.edu>

Examples

```
res <- ci_smd(smd = 0.5, n_1 = 50, n_2 = 50)
generics::tidy(res)
generics::glance(res)
```

`tidy.dmar_contrast_test` *Tidy and Glance Methods for DMAR Result Tables*

Description

A DMAR function returns a tidy `data.frame` built to be read: one row per quantity, a numeric `value` column, and a display layer that rounds sensibly on the way to the console (see `dmar_tbl`). The tidy verbs `tidy` and `glance` give the same numbers in the two shapes a programmer usually wants instead: one row per term with a typed column for each quantity, and a one-row summary of the result as a whole. This page states that contract once, for the confidence interval family, the post hoc family, the contrast tests, and the power-based sample size planners.

Usage

```
## S3 method for class 'dmar_contrast_test'
tidy(x, ...)

## S3 method for class 'dmar_contrast_test'
glance(x, ...)

## S3 method for class 'dmar_ci_long'
tidy(x, ...)

## S3 method for class 'dmar_ci_long'
glance(x, ...)

## S3 method for class 'dmar_ci_anova'
tidy(x, ...)

## S3 method for class 'dmar_ci_anova'
glance(x, ...)

## S3 method for class 'dmar_post_hoc_ci'
tidy(x, ...)

## S3 method for class 'dmar_post_hoc_ci'
glance(x, ...)

## S3 method for class 'dmar_ss_power'
tidy(x, ...)

## S3 method for class 'dmar_ss_power'
glance(x, ...)

## S3 method for class 'dmar_ss_aipe'
tidy(x, ...)
```

```
## S3 method for class 'dmar_ss_aipe'
glance(x, ...)

## S3 method for class 'dmar_tbl'
tidy(x, ...)

## S3 method for class 'dmar_tbl'
glance(x, ...)

## S3 method for class 'dmar_content_validity'
tidy(x, ...)

## S3 method for class 'dmar_content_validity'
glance(x, ...)

## S3 method for class 'dmar_dmacs'
tidy(x, ...)

## S3 method for class 'dmar_dmacs'
glance(x, ...)

## S3 method for class 'dmar_measurement_invariance'
tidy(x, ...)

## S3 method for class 'dmar_measurement_invariance'
glance(x, ...)

## S3 method for class 'dmar_measurement_alignment'
tidy(x, ...)

## S3 method for class 'dmar_measurement_alignment'
glance(x, ...)

## S3 method for class 'dmar_ss_power_sensitivity'
tidy(x, ...)

## S3 method for class 'dmar_ss_power_sensitivity'
glance(x, ...)
```

Arguments

x	A DMAR result object carrying one of the classes listed above.
...	Unused, present for consistency with the generics.

Details

What the verbs return. `tidy(x)` returns a `data.frame` with one row per term, where a term is whatever the family produces one of: a parameter estimate, a contrast, a planned design. Its columns

follow the naming convention the broom ecosystem uses, which separates words with dots rather than the underscores DMAR uses everywhere else: `term`, `estimate`, `se`, `statistic`, `p_value`, `ci_lower`, `ci_upper`, and `conf_level`. A method reports the subset of those columns its family can fill, plus any column the family genuinely adds, such as `p_adjusted` for a multiplicity-adjusted set of comparisons or `power` for a sample size planner.

`glance(x)` returns a one-row `data.frame` summarizing the result as a whole, in the same dotted convention: how many comparisons were made, at what confidence level, with which planning inputs. When a result has a single estimand and nothing further to say at the model level, as for a lone effect size and its confidence interval, `glance()` coincides with `tidy()`. That is expected rather than a defect, since there is no model-level quantity that the single row does not already carry.

Neither verb rounds. The `dmar_tbl` layer formats what is printed, while `tidy()` and `glance()` return full precision, which is what makes them the right input to a downstream calculation or plot.

Why broom is not a dependency. The `tidy()` and `glance()` generics live in **generics**, a small package that holds the generics and little else. **broom** imports them from there, and so does DMAR, which registers its methods against `generics::tidy` and `generics::glance` rather than against **broom** itself. A user with **broom** or the tidymodels stack loaded gets DMAR methods on the generic they already call; a user with neither installed can still call `generics::tidy()` directly. DMAR never loads **broom**, and does not need it installed.

The families and the classes they carry. Each family tags its return with a leading S3 class, ahead of `dmar_tbl` and `data.frame`, so the verbs dispatch while printing and data-frame behavior are untouched.

S3 class	Family	One `tidy()` row is
`dmar_ci_long`	confidence intervals, long form	an estimate and its limits
`dmar_ci_anova`	ANOVA effect size intervals	an effect size and its limits
`dmar_post_hoc_ci`	simultaneous intervals	one pairwise or one contrast comparison
`dmar_contrast_test`	contrast tests	one contrast, with its test and its interval
`dmar_ss_power`	sample size planners	a planned size and the power it buys
`dmar_ss_power_sensitivity`	planner sensitivity studies	a planned size and two powers

The confidence interval family. Two classes cover the two output shapes.

`dmar_ci_long` Long-format interval tables, with rows for `lower_limit` and `upper_limit` and, when the function reports one, an estimate row whose `term` is the name of the parameter. Carried by `ci_r`, `ci_smd_c`, `ci_pvaf`, and `ci_reg_coef`.

`dmar_ci_anova` Wide-format ANOVA effect size interval tables, with one row and columns for the effect name, the point estimate, the limits, and the design metadata. Carried by `ci_eta_squared`, `ci_eta_squared_partial`, `ci_eta_squared_generalized`, and `ci_omega_squared`.

Both produce a one-row `data.frame` with `term`, `estimate`, `ci_lower`, `ci_upper`, and, when the object records it, `conf_level`. `glance()` on either class calls `tidy()`, since the row is already the whole result.

The post hoc family. `ci_tukey_kramer`, `ci_games_howell`, `ci_scheffe`, and `ci_dunnett` all carry `dmar_post_hoc_ci`. Their source table is wide, with one row per comparison: a `contrast` label, a point estimate (`mean_difference` for the pairwise and many-to-one procedures, `contrast_value` for Scheffe), a standard error, a test statistic, the `lower_limit` and `upper_limit` of the simultaneous interval, and the multiplicity-adjusted `p_adjusted`. `tidy()` maps that to `term`, `estimate`,

`ci_lower`, `ci_upper`, `p_adjusted`, and `conf_level`, one row per comparison. `glance()` describes the family of comparisons as a whole: how many there were, and the simultaneous confidence level they hold jointly.

The contrast tests. `contrast_test` carries `dmar_contrast_test`. Its source table is wide, with one row per contrast: a `contrast` label, the estimate $\hat{\psi} = \sum_i c_i \bar{Y}_i$, its standard error, the *t*-statistic and the degrees of freedom it is referred to, the unadjusted *p*-value, the multiplicity-adjusted `p_adjusted`, and the `ci_lower` and `ci_upper` limits. `tidy()` renames those to `term`, `estimate`, `ci_lower`, `ci_upper`, `statistic`, `df`, `p_value`, `p_adjusted`, and `conf_level`, one row per contrast. Both *p*-values are kept, because the pair is what a contrast table is read for: what the contrast would show on its own, and what it shows once the family it belongs to is accounted for.

Where a post hoc procedure fixes its adjustment as part of the method, a contrast test chooses one, and the same weights tested under `adjust = "none"` and under `adjust = "tukey"` are two different inferences. `glance()` therefore records the choice alongside the family-level numbers: `n_contrasts`, `adjust`, `var_equal`, the smallest adjusted *p*-value `p_adjusted_min`, and `conf_level`. `adjust` and `var_equal` name a procedure rather than measure a quantity, so this one-row summary, unlike a DMAR result table, is not numeric throughout.

The power-based sample size planners. A planner in the `ss_power_*` family returns a long table with a row for the recommended sample size, a row for the realized power, and rows echoing the planning inputs. A planner that reports one size and one power for one design tags its return `dmar_ss_power`. This covers the closed-form effect size planners (`ss_power_R2`, `ss_power_r`, `ss_power_reg_coef`, `ss_power_smd`, `ss_power_sem`), the contrast and ANCOVA planners (`ss_power_c`, `ss_power_c_ancova`, `ss_power_sc`, `ss_power_contrast`, `ss_power_equivalence_c`), the ANOVA and cluster designs (`ss_power_one_way_anova`, `ss_power_factorial_anova`, `ss_power_factorial_ancova`, `ss_power_split_plot_anova`, `ss_power_rm_anova`, `ss_power_mixed_effects`), and the mediation planner `ss_power_indirect_effect`, whose reported power is the joint power to detect the indirect effect and whose component path powers `glance()` carries as extra columns.

The size `tidy()` reports is the design's planning unit: per group, per cell, per subject, or per cluster. The one-way ANOVA planner, whose natural unit is the total, is summarized by its total N. A design that reports two group sizes reports one of them beside the realized power, falling through to the total N when the per-group sizes are unequal, and `glance()` keeps every group size as a column so none is lost. A planner whose result spans several effects, with no single size-and-power summary to give, returns a plain `dmar_tbl` and does not gain these verbs at all.

The Monte Carlo sensitivity siblings `ss_power_R2_sensitivity` and `ss_power_reg_coef_sensitivity` report two powers at one planned sample size, the empirical (simulated) power and the analytic power, and comparing the two is the object of the study. They carry `dmar_ss_power_sensitivity` instead: `tidy()` places both powers beside the planned `sample_size`, and `glance()` adds the simulated distribution of the estimator.

Adding a planner to the family. A planner opts in by setting `dmar_ss_power` as a leading class before routing its return through `.as_dmar_tbl()`. The rows the verbs read are named in the internal vectors `.SS_POWER_SIZE_TERMS` and `.SS_POWER_POWER_TERMS`. A planner whose size or power row is not named there reports `NA` rather than failing, so a new row name has to be added to those vectors when a planner introduces one.

Value

`tidy()` returns a `data.frame` with one row per term and broom-convention column names. `glance()`

returns a one-row `data.frame` summarizing the result as a whole. Both return values at full precision.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

`dmar_tbl` for the printing layer these tables share, and the "Reading DMAR result tables" vignette for the wider output convention.

Examples

```
# A single interval: tidy() and glance() coincide, because there is
# nothing at the model level the one row does not already carry.
res <- ci_r(r = 0.5, n = 50)
generics::tidy(res)
generics::glance(res)

# A family of simultaneous intervals: one tidy() row per comparison,
# one glance() row describing the family.
set.seed(113)
y <- c(rnorm(10, 0), rnorm(10, 1), rnorm(10, 2))
g <- factor(rep(c("a", "b", "c"), each = 10))
gh <- ci_games_howell(y, group = g)
generics::tidy(gh)
generics::glance(gh)

# A set of contrasts: tidy() keeps both the unadjusted and the
# adjusted p-value, and glance() names the adjustment that produced
# the second of them.
fit <- aov(bdi_post ~ condition, data = depression_bdi)
ct <- contrast_test(fit, contrasts = "pairwise", adjust = "tukey")
generics::tidy(ct)
generics::glance(ct)

# A sample size planner: tidy() gives the size and the power it buys,
# glance() adds the planning inputs that produced them.
plan <- ss_power_smd(smd = 0.5, desired_power = 0.80)
generics::tidy(plan)
generics::glance(plan)
```

`tidy.dmar_mediation_mbco`

Tidy an MBCO Mediation Table

Description

Returns the effect rows of a mediation_mbco table in the column convention used by the **broom** ecosystem. The statistic column is the MBCO likelihood ratio statistic.

Usage

```
## S3 method for class 'dmar_mediation_mbco'
tidy(x, ...)
```

Arguments

x	A dmar_mediation_mbco object returned by mediation_mbco.
...	Unused.

Value

A data.frame with columns term, estimate, se, statistic, p_value, ci_lower, ci_upper.

Author(s)

Ken Kelley <kkelley@nd.edu>

---

tidy.dmar_R2_mixed_effects

Tidy / Glance Methods for R2_mixed_effects Output

Description

Returns the broom-style summary of the marginal and conditional R^2: term ("R2_marginal" or "R2_conditional"), estimate, and, when a bootstrap interval was requested, ci_lower, ci_upper, and conf_level. glance returns the same information in one row per quantity.

Usage

```
## S3 method for class 'dmar_R2_mixed_effects'
tidy(x, ...)

## S3 method for class 'dmar_R2_mixed_effects'
glance(x, ...)
```

Arguments

x	A dmar_R2_mixed_effects object returned by R2_mixed_effects.
...	Unused.

Value

A data.frame in broom convention.

Author(s)

Ken Kelley <kkelley@nd.edu>

Examples

```
fit <- lme4::lmer(Reaction ~ Days + (Days | Subject),
                  data = lme4::sleepstudy)
res <- R2_mixed_effects(fit)
generics::tidy(res)
generics::glance(res)
```

---

tidy.dmar_reliability *A Reliability Coefficient Estimate*

Description

Returns a one-row `data.frame` in the column convention used by the **broom** ecosystem (`term`, `estimate`, `se`, `ci_lower`, `ci_upper`). The `term` is the coefficient name (`"alpha"`, `"omega"`, etc.).

Usage

```
## S3 method for class 'dmar_reliability'
tidy(x, ...)
```

Arguments

`x`	A `dmar_reliability` object returned by any of `reliability_alpha`, `reliability_kr20`, `reliability_omega`, `reliability_omega_categorical`, or `reliability`.
`...`	Unused.

Value

A one-row `data.frame`.

Author(s)

Ken Kelley <kkelley@nd.edu>

Examples

```
# Coefficient alpha for the three verbal tests of the Holzinger and
# Swineford battery, from their covariance matrix.
S <- cov(holzinger_swineford[, c("t6_paragraph_comprehension",
                                 "t7_sentence", "t9_word_meaning")])
res <- reliability_alpha(S = S, N = 301, ci_method = "feldt")
generics::tidy(res)
generics::glance(res)
```

tidy.mlmr *An Mlmr Fit*

Description

Returns a one-row-per-coefficient `data.frame` in the column convention used by the **broom** ecosystem (`term`, `estimate`, `se`, `statistic`, `p_value`, and optionally `ci_lower`, `ci_upper`). Use `as.data.frame()` on the fit for the DMAR-style table (snake_case columns) stored at `fit$coef_table`.

Usage

```
## S3 method for class 'mlmr'
tidy(x, conf.int = FALSE, conf_level = NULL, standardized = FALSE, ...)
```

Arguments

`x`	An object of class `"mlmr"`.
`conf.int`	Logical; if `TRUE`, append `ci_lower` and `ci_upper` columns using the confidence intervals already computed at fit time. Defaults to `FALSE`.
`conf_level`	Ignored; the confidence interval comes from the fit object at `x$conf_level`. Present for compatibility with the broom generic.
`standardized`	Logical; if `TRUE` and effect sizes were computed at fit time, append a `std_estimate` column. Defaults to `FALSE`.
`...`	Unused.

Value

A `data.frame`.

Author(s)

Ken Kelley <kkelley@nd.edu>

Examples

```
fit <- mlmr(t6_paragraph_comprehension ~ t5_general_information +
              t9_word_meaning,
            data = holzinger_swineford, ci_method = "wald")
generics::tidy(fit)
generics::tidy(fit, conf.int = TRUE)
generics::tidy(fit, conf.int = TRUE, standardized = TRUE)
```

`tidy.mlmr_mv` *A Multivariate FIML Regression Fit*

Description

Broom-style `tidy()` and `glance()` for `mlmr_mv` fits. `tidy()` returns one row per coefficient per outcome with the broom dotted columns plus a leading `response` column identifying the outcome; `glance()` returns a one-row model-level summary with the per-outcome R^2 averaged and the number of responses reported.

Usage

```
## S3 method for class 'mlmr_mv'
tidy(x, conf.int = FALSE, conf_level = NULL, standardized = FALSE, ...)

## S3 method for class 'mlmr_mv'
glance(x, ...)
```

Arguments

`x`	An `mlmr_mv` fit.
`conf.int`	Logical: include `ci_lower` / `ci_upper`?
`conf_level`	Ignored (the interval level is fixed at fit time and stored on the object); present for broom signature compatibility.
`standardized`	Logical: include `std_estimate`?
`...`	Unused.

Value

For `tidy.mlmr_mv`, a `data.frame` with columns `response`, `term`, `estimate`, `se`, `statistic`, `p_value`, and optionally `ci_lower`, `ci_upper`, `std_estimate`. For `glance.mlmr_mv`, a one-row `data.frame` with `R2` (mean across outcomes), `df`, `logLik`, `AIC`, `BIC`, `deviance`, `nobs`, and `n.responses`.

Author(s)

Ken Kelley <kkelley@nd.edu>

See Also

`mlmr_mv`; `mlmr` for the univariate methods these mirror.

`unbiased_R2` *Unbiased and Adjusted Estimators of the Population Squared Multiple Correlation*

Description

Estimates the population squared multiple correlation coefficient ρ^2 from an observed sample R^2, correcting the well-known positive (upward) bias of R^2. Two estimators are available: the (essentially) unbiased Olkin and Pratt (1958) estimator (the default), and the classic Ezekiel (1930) adjusted-R^2 shrinkage formula reported by `summary.lm` as `adj.r.squared`. This is the inverse direction of `expected_R2`, which gives the forward expectation $E[R^2 \mid \rho^2]$.

Usage

```
unbiased_R2(R2, N, p, method = c("olkin_pratt", "ezekiel"))
```

Arguments

`R2`	Observed sample squared multiple correlation coefficient, in $[0, 1]$.
`N`	Sample size.
`p`	Number of predictor variables.
`method`	Which estimator to compute: `"olkin_pratt"` (default) for the (essentially) unbiased Olkin-Pratt (1958) estimator, or `"ezekiel"` for the Ezekiel (1930) adjusted R^2 (the value `summary.lm` reports as `adj.r.squared`).

Details

The sample R^2 overestimates ρ^2; the bias is larger for smaller samples and for more predictors. Two corrections are offered.

The Ezekiel (1930) adjusted estimator is

$$\hat{\rho}^2_{\text{Ezekiel}} = 1 - \frac{N-1}{N-p-1}\,(1 - R^2).$$

It reduces the bias but is not unbiased; it is exactly the quantity `summary(lm(...))$adj.r.squared` reports.

The Olkin and Pratt (1958) estimator is (essentially) unbiased:

$$\hat{\rho}^2_{\text{OP}} = 1 - \frac{N-3}{N-p-1}\,(1 - R^2)\;{}_2F_1\left(1, 1; \frac{N-p+1}{2}; 1 - R^2\right),$$

where ${}_2F_1$ is the Gaussian hypergeometric function (the same function `expected_R2` uses for the forward direction; see Stuart, Ord, & Arnold, 1999, section 28). Both estimators can fall below 0 for very small R^2; that is expected behavior for a bias-corrected estimator and is not truncated here (matching `adj.r.squared`, which is also allowed to be negative).

Value

A 1-row `data.frame` (class `dmar_tbl`) with columns `term` and `value`. The `term` is `"unbiased_population_R2"` when `method = "olkin_pratt"` and `"adjusted_population_R2"` when `method = "ezekiel"`; `value` is the corresponding estimate of ρ^2.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Ezekiel, M. (1930). *Methods of correlation analysis*. Wiley.

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*(4), 524–555. doi:10.1080/00273170802490632

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on R^2 as a model comparison effect size.)

Olkin, I., & Pratt, J. W. (1958). Unbiased estimation of certain correlation coefficients. *The Annals of Mathematical Statistics, 29*(1), 201–211.

Stuart, A., Ord, J. K., & Arnold, S. (1999). *Kendall's advanced theory of statistics, volume 2A: Classical inference and the linear model* (6th ed.). Arnold.

See Also

`expected_R2` for the forward expectation $E[R^2 \mid \rho^2]$, and `ci_R2`, `var_R2`, `ss_aipe_R2` for interval, variance, and planning tools on the same effect size.

Examples

```
# An observed R^2 = .50 with N = 50 and p = 5 predictors overstates rho^2.
# The Olkin-Pratt (essentially unbiased) estimate is the default:
unbiased_R2(R2 = .50, N = 50, p = 5)

# The Ezekiel adjusted R^2 (what summary(lm) reports) over-shrinks slightly,
# so it typically sits a little below the Olkin-Pratt value:
unbiased_R2(R2 = .50, N = 50, p = 5, method = "ezekiel")

# The Ezekiel option reproduces summary(lm)$adj.r.squared exactly.
set.seed(113)
d   <- as.data.frame(matrix(rnorm(50 * 6), 50, 6))
fit <- lm(V1 ~ ., data = d)
s   <- summary(fit)
unbiased_R2(R2 = s$r.squared, N = 50, p = 5, method = "ezekiel")$value
s$adj.r.squared

# The bias (and so the correction) shrinks as N grows for fixed R^2 and p.
unbiased_R2(.50, 50, 5)
unbiased_R2(.50, 500, 5)
```

var_alpha *Asymptotic Variance of Coefficient Alpha (Cronbach, Guttman)*

Description

Computes the asymptotic variance of the sample coefficient alpha (Guttman, 1945; Cronbach, 1951) under multivariate normality of the p item scores, using the closed form derived by van Zyl, Neudecker, & Nel (2000) under the assumption that the population covariance matrix has equal off-diagonals (the parallel-items model), along with the simpler Bonett (2002) approximation $2p/(p-1) \cdot (1-\alpha)^2/(n-1)$ that is widely used in planning.

Usage

```
var_alpha(alpha, n, p_items)
```

Arguments

alpha	Population coefficient alpha. Numeric scalar in $[0, 1)$.
n	Sample size (number of *respondents*).
p_items	Number of items contributing to the composite alpha coefficient.

Details

Companion to the existing reliability-coefficient infrastructure (reliability_alpha, reliability) and a building block for AIPE planning around alpha.

van Zyl-Neudecker-Nel (2000) variance. Under multivariate normality and the parallel-items model (all items have equal variances and equal pairwise covariances), the asymptotic variance of the maximum likelihood estimator of alpha is

$$\mathrm{Var}(\hat{\alpha}) \;=\; \frac{2p(1-\alpha)^2}{(p-1)(n-2)}.$$

This is one of two closed forms van Zyl et al. derive; the more general (non-parallel) form involves matrix expressions and is implemented separately by the existing reliability infrastructure.

Bonett (2002) approximation. Bonett (2002) gives the easy planning form

$$\mathrm{Var}(\hat{\alpha}) \;\approx\; \frac{2p}{(p-1)} \cdot \frac{(1-\alpha)^2}{n-1}.$$

This differs from the van Zyl form only in the denominator ($n-1$ vs.\ $n-2$) and converges to the same value for moderate n. Bonett's version is what most sample size tables use.

When to use which. For inference (a CI on α), the van Zyl form is preferable, especially at small n; for sample size *planning* the difference is immaterial and the Bonett form is widely cited and easier to invert.

Value

A `data.frame` with rows for the van Zyl, Neudecker, & Nel (2000) exact-under-parallel-items variance and the Bonett (2002) simpler approximation; columns are `term` and `value`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bonett, D. G. (2002). Sample size requirements for testing and estimating coefficient alpha. *Journal of Educational and Behavioral Statistics, 27*(4), 335–340. doi:10.3102/10769986027004335

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika, 16*(3), 297–334.

Guttman, L. (1945). A basis for analyzing test-retest reliability. *Psychometrika, 10*(4), 255–282.

Kelley, K., & Cheng, Y. (2012). Estimation of and confidence interval formation for reliability coefficients of homogeneous measurement instruments. *Methodology, 8*, 39–50. doi:10.1027/1614-2241/a000036

Kelley, K., & Pornprasertmanit, S. (2016). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for composite measures. *Psychological Methods, 21*, 69–92. doi:10.1037/a0040086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

McDonald, R. P. (1999). *Test theory: A unified treatment*. Lawrence Erlbaum.

Terry, L. J., & Kelley, K. (2012). Sample size planning for composite reliability coefficients: Accuracy in parameter estimation via narrow confidence intervals. *British Journal of Mathematical and Statistical Psychology, 65*, 371–401. doi:10.1111/j.20448317.2011.02030.x

van Zyl, J. M., Neudecker, H., & Nel, D. G. (2000). On the distribution of the maximum likelihood estimator of Cronbach's alpha. *Psychometrika, 65*(3), 271–280. doi:10.1007/BF02296146

See Also

`reliability_alpha`, `reliability`, `ss_aipe_reliability`

Other variance utilities: `var_cv()`, `var_ete()`, `var_indirect_effect()`, `var_omega_squared()`, `var_r()`, `var_smd()`, `var_smd_trimmed()`

Examples

```
# 1. Variance of alpha = 0.80 from a 10-item test with n = 100.
var_alpha(alpha = 0.80, n = 100, p_items = 10)

# 2. Variance shrinks with n and grows as alpha moves away from 1:
var_alpha(alpha = 0.90, n = 50,  p_items = 5)
var_alpha(alpha = 0.90, n = 500, p_items = 5)
```

var_cv *Asymptotic Variance of the Coefficient of Variation*

Description

Computes the asymptotic variance of the sample coefficient of variation $\hat{\kappa} = s/\bar{Y}$ under normality, using McKay's (1932) original noncentral t-based approximation and Vangel's (1996) refinement. Companion to `ci_cv` and `ss_aipe_cv`.

Usage

```
var_cv(cv, n)
```

Arguments

`cv`	Population coefficient of variation $\kappa = \sigma/\mu$. Numeric scalar in $(0, \infty)$. When κ is very large (> 0.5), both McKay's and Vangel's approximations degrade and exact methods (`ci_cv`) should be preferred.
`n`	Sample size.

Details

McKay (1932). The classical large-sample variance of the sample CV under normality is

$$\mathrm{Var}(\hat{\kappa}) \;\approx\; \frac{\kappa^2}{n-1} \cdot \left(\frac{1}{2} + \kappa^2 \right).$$

This is exact up to $O(1/n)$ and is what most planning tables use. It begins to drift when $\kappa > 0.3$ or so.

Vangel (1996). Vangel showed that a small-sample correction that adjusts the McKay form for the noncentral t mean factor gives substantially better coverage of CIs derived from the variance:

$$\mathrm{Var}_{\mathrm{Vangel}}(\hat{\kappa}) \;\approx\; \frac{\kappa^2}{n-1} \cdot \left(\frac{1}{2} + \kappa^2 \cdot \frac{n+1}{n-1} \right).$$

The two forms coincide in the large-n limit. We report both so the user can see the magnitude of the small-sample correction.

Value

A `data.frame` with rows for the McKay (1932) and Vangel (1996) approximations; columns are `term` and `value`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007). Sample size planning for the coefficient of variation from the accuracy in parameter estimation approach. *Behavior Research Methods, 39*(4), 755–766. doi:10.3758/BF03192966

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3.)

McKay, A. T. (1932). Distribution of the coefficient of variation and the extended *t* distribution. *Journal of the Royal Statistical Society, 95*(4), 695–698.

Vangel, M. G. (1996). Confidence intervals for a normal coefficient of variation. *The American Statistician, 50*(1), 21–26. doi:10.1080/00031305.1996.10473537

See Also

`ci_cv`, `ss_aipe_cv`, `ss_aipe_cv_sensitivity`

Other variance utilities: `var_alpha()`, `var_ete()`, `var_indirect_effect()`, `var_omega_squared()`, `var_r()`, `var_smd()`, `var_smd_trimmed()`

Examples

```
# 1. CV = 0.20 in a sample of 30:
var_cv(cv = 0.20, n = 30)

# 2. The Vangel correction grows with cv (becomes non-trivial
#        for kappa > 0.3):
var_cv(cv = 0.50, n = 30)
```

---

var_ete *Variance of the Estimated Treatment Effect in Two-Group ANCOVA With Heterogeneous Slopes*

Description

Computes the variance of the estimated treatment effect (ETE) at a chosen covariate value in a two-group analysis of covariance with heterogeneity of regression and a random covariate, following Li, McLouth, and Delaney (2020). When the two groups' slopes differ, the treatment effect is a function of the covariate, and its sampling variance at the sample grand mean, one standard deviation from the mean, or a fixed covariate value must account for the covariate being a random variable rather than a set of fixed constants; the fixed-constant formulas understate or misstate that variability. This is the reimplementation of `var.ete()` from **MBESS**, contributed there by Li Li.

Usage

```
var_ete(
  sigma2,
  sigma2_Z,
  n_1,
```

```
  n_2,
  beta_1,
  beta_2,
  mu_Z = 0,
  fixed_value = 0,
  type = c("sample", "population"),
  covariate_value = c("sample_mean", "sd", "fixed")
)
```

Arguments

`sigma2`	Residual error variance: the population value when `type = "population"`, the sample estimate when `type = "sample"`.
`sigma2_Z`	Variance of the random covariate: population value or sample estimate, matching `type`.
`n_1`, `n_2`	Sample sizes of the two groups (each must exceed 3; the formulas involve $n - 3$ in denominators).
`beta_1`, `beta_2`	Slopes of the covariate in group 1 and group 2: population values or sample estimates, matching `type`.
`mu_Z`	Mean of the covariate (population value or sample mean, matching `type`). Defaults to 0. Used when `covariate_value = "fixed"`.
`fixed_value`	The fixed covariate value at which the treatment effect is assessed when `covariate_value = "fixed"`. Defaults to 0.
`type`	`"sample"` (default) for the unbiased estimate of the variance from sample slopes and variances, or `"population"` for the variance from population values.
`covariate_value`	Where the treatment effect is assessed: `"sample_mean"` (default) at the sample grand mean of the covariate, `"sd"` at the grand mean plus or minus one sample standard deviation, or `"fixed"` at `fixed_value`.

Details

Randomized experiments with a covariate commonly probe the simple treatment effect at the grand mean and one standard deviation either side of it when the slopes differ across groups. The variance expressions here treat the covariate as normally distributed rather than fixed, which Li, McLouth, and Delaney (2020) show can change the estimated standard error substantially when heterogeneity of regression is strong. The square root of the returned value is the standard error used for a confidence interval or test of the treatment effect at the chosen covariate value.

At the sample grand mean of the covariate, writing $N = n_1 + n_2$, the population variance (their Equation 10) is

$$\mathrm{Var} = \sigma^2 C_0 + \frac{(\beta_1 - \beta_2)^2 \sigma_Z^2}{N}, \qquad C_0 = \frac{1}{n_1} + \frac{1}{n_2} + \frac{n_2}{N n_1 (n_1 - 3)} + \frac{n_1}{N n_2 (n_2 - 3)},$$

and with `type = "sample"` the returned value is their unbiased estimator (Equation C.7), which subtracts $\sigma^2\{(N-3)/(n_1-3)+(N-3)/(n_2-3)\}/\{N(N-1)\}$ so that plugging in sample estimates does not overstate the variance. The `covariate_value = "sd"` expressions are their Equations 12

and C.9, which add the variance contribution of estimating the covariate's standard deviation, and the `"fixed"` expressions are their Equations 14 and C.10.

The two `"fixed"` estimands differ in where the deviation of `fixed_value` is measured from. With `type = "sample"` the deviation is taken from the sample grand mean (Equation C.10), so `mu_Z` should be the sample mean of the covariate. With `type = "population"` the deviation is taken from a known population mean (Equation 14); evaluating the treatment effect at that known mean itself, as in the paper's worked example, sets `fixed_value = mu_Z`, which zeroes the deviation term.

Value

A `data.frame` (class `dmar_tbl`) with one row, `term = "var_ete"`, whose `value` is the variance of the estimated treatment effect at the chosen covariate value. The `type` and `covariate_value` choices are recorded as attributes of the same names.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2007a). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K. (2007b). Methods for the behavioral, educational, and social sciences: An R package. *Behavior Research Methods, 39*(4), 979–984. doi:10.3758/BF03192993

Li, L., McLouth, C. J., & Delaney, H. D. (2020). Analysis of covariance in randomized experiments with heterogeneity of regression and a random covariate: The variance of the estimated treatment effect at selected covariate values. *Multivariate Behavioral Research, 55*(6), 926–940. doi:10.1080/00273171.2019.1693953

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 9 on heterogeneity of regression.)

See Also

`ancova` for the model whose treatment effect this variance describes; `regions_of_significance` for the companion question of where a moderated effect is distinguishable from zero.

Other variance utilities: `var_alpha()`, `var_cv()`, `var_indirect_effect()`, `var_omega_squared()`, `var_r()`, `var_smd()`, `var_smd_trimmed()`

Examples

```
# Pygmalion data (Maxwell, Delaney, & Kelley, 2027): the treatment
# effect of the "Bloomer" expectation at the covariate grand mean,
# with heterogeneous pre-IQ slopes.
data(pygmalion)
fit <- lm(iq_8 ~ iq_pre * treatment, data = pygmalion)
s2  <- sum(residuals(fit)^2) / fit$df.residual
var_ete(sigma2 = s2, sigma2_Z = var(pygmalion$iq_pre),
        n_1 = sum(pygmalion$treatment == "Bloomer"),
        n_2 = sum(pygmalion$treatment == "Control"),
```

```
          beta_1 = coef(fit)["iq_pre"] + coef(fit)["iq_pre:treatmentBloomer"],
          beta_2 = coef(fit)["iq_pre"])
```

`var_icc` *Asymptotic Variance of the Intraclass Correlation Coefficient*

Description

Computes the asymptotic (large-sample) variance of the intraclass correlation coefficient (ICC) for any of the six classical Shrout-Fleiss (1979) forms, given a value of the population ICC, the number of subjects n, and the number of raters k. The single-rater forms use Smith's (1956) one-way-ANOVA asymptotic variance, equivalently the Fisher (1925) information-matrix result, and the average-of-k forms use a delta method transformation through the Spearman-Brown relation.

Usage

```
var_icc(rho, n, k, type = "ICC(1,1)")
```

Arguments

`rho`	The intraclass correlation coefficient at which the asymptotic variance is evaluated, on the scale matching `type`: for the single-rater types (`"ICC(1,1)"`, `"ICC(2,1)"`, `"ICC(3,1)"`) supply the single-rater ICC; for the average-of-k types (`"ICC(1,k)"`, `"ICC(2,k)"`, `"ICC(3,k)"`) supply the average-of-k ICC. Must lie in $[0, 1]$. **Population value or sample value?** The formula is derived in terms of the unknown population value ρ. In applied work ρ is never known, so two conventions are common: (i) for *prospective* variance calculation (e.g., when planning a study and asking how precise an estimator will be at plausible truths), supply your *anticipated* population value motivated by prior literature, theory, or a pilot; (ii) for *post hoc* variance calculation (e.g., constructing a Wald standard error or a meta-analytic weight from an observed sample), supply the *sample* estimate $\hat{\rho}$ as a plug-in for the population value. The two postures look identical at the call site but have different conceptual content; the plug-in case (ii) yields a consistent (not exact) estimator of the variance.
`n`	Number of subjects (targets) rated.
`k`	Number of raters (or repeated measurements) per subject.
`type`	Which Shrout-Fleiss ICC variant the population value represents. One of `"ICC(1,1)"`, `"ICC(2,1)"`, `"ICC(3,1)"`, `"ICC(1,k)"`, `"ICC(2,k)"`, `"ICC(3,k)"`. The shorthand aliases used in `icc` (`"1"`, `"2"`, `"3"`, `"1k"`, `"2k"`, `"3k"`) are also accepted.

Details

Single-rater forms. For the single-rater intraclass correlation in a balanced design with n subjects and k measurements per subject, the standard large-sample variance derived from the Fisher information matrix is (Smith, 1956; Donner, 1986; Searle, 1971, ch. 11):

$$\mathrm{Var}(\hat{\rho}) \;\approx\; \frac{2(1-\rho)^2\big(1+(k-1)\rho\big)^2}{n\,k\,(k-1)}.$$

This expression is exact under the one-way random-effects model (`ICC(1,1)`) and serves as the standard large-sample approximation for the two-way single-rater forms (`ICC(2,1)`, `ICC(3,1)`) as well; differences with the exact two-way variance vanish at order $1/n$ (Bonett, 2002; Burdick & Graybill, 1992). For small n or moderate k where exact two-way inference matters, prefer the F-distribution-based confidence intervals returned by `icc`, which follow Shrout and Fleiss (1979) directly.

Average-of-*k* forms. Applying the Spearman-Brown transformation $\rho_k = k\rho/[1 + (k-1)\rho]$ together with the delta method gives the closed-form

$$\mathrm{Var}(\hat{\rho}_k) \;\approx\; \frac{2\,k\,(1-\rho_k)^2}{n\,(k-1)},$$

expressed directly in the average-level ICC ρ_k (so the user need not invert Spearman-Brown when working with reliability of composites). The reduction to this form follows from the substitutions $1-\rho = k(1-\rho_k)/[k-(k-1)\rho_k]$ and $1+(k-1)\rho = k/[k-(k-1)\rho_k]$.

Use cases. The asymptotic variance is the natural ingredient for Wald-style inference, sample size planning for the width of an ICC confidence interval (compare with Bonett, 2002, which uses a Fisher-style transformation), and meta-analytic synthesis of ICCs across studies (the inverse of `value` weights each study). For confidence intervals themselves, prefer `icc`, which uses the exact F-distribution inversion of Shrout and Fleiss (1979, pp. 425–426).

Value

A one-row `data.frame` with columns `term` (always `"var_icc"`) and `value` (the asymptotic variance).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bonett, D. G. (2002). Sample size requirements for estimating intraclass correlations with desired precision. *Statistics in Medicine, 21*(9), 1331–1335. doi:10.1002/sim.1108

Burdick, R. K., & Graybill, F. A. (1992). *Confidence Intervals on Variance Components*. Marcel Dekker.

Donner, A. (1986). A review of inference procedures for the intraclass correlation coefficient in the one-way random effects model. *International Statistical Review, 54*(1), 67–82.

Fisher, R. A. (1925). *Statistical Methods for Research Workers*. Oliver & Boyd.

McGraw, K. O., & Wong, S. P. (1996). Forming inferences about some intraclass correlation coefficients. *Psychological Methods, 1*(1), 30–46. doi:10.1037/1082989X.1.1.30

Searle, S. R. (1971). *Linear Models*. Wiley.

Shrout, P. E., & Fleiss, J. L. (1979). Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin, 86*(2), 420–428.

Smith, C. A. B. (1956). On the estimation of intraclass correlation. *Annals of Human Genetics, 21*(4), 363–373.

See Also

icc, ss_aipe_reliability, var_R2

Examples

```
# Single-rater one-way ICC at rho = .60 with 30 subjects and 4 raters.
var_icc(rho = 0.60, n = 30, k = 4, type = "ICC(1,1)")

# Same study, but expressed at the average-of-4-rater level. The
# Spearman-Brown relation carries the single-rater value of .60 up
# to the average-of-4 scale, about .857.
rho_k <- 4 * 0.60 / (1 + 3 * 0.60)
rho_k
var_icc(rho = rho_k, n = 30, k = 4, type = "ICC(1,k)")

# Two-way mixed-model consistency ICC, which uses the same one-way
# asymptotic variance as a large-sample approximation (see Details).
var_icc(rho = 0.60, n = 30, k = 4, type = "ICC(3,1)")
```

---

var_indirect_effect *Variance of the Mediated (Indirect) Effect ab*

Description

Computes the asymptotic variance of the product of two regression coefficients $\hat{a}\hat{b}$ (the mediated/indirect effect in a simple three-variable mediator model: $X \rightarrow M \rightarrow Y$) under four competing formulas: Sobel (1982) first-order, Aroian (1947) / Goodman (1960) second-order, and the full second-order delta method with optional cross-product covariance. All four are reported in a single output so the user can see the relative contributions of the higher-order terms.

Usage

```
var_indirect_effect(a, b, var_a, var_b, cov_ab = 0)
```

Arguments

a, b	Anticipated population (or estimated) regression coefficients for $X \rightarrow M$ and $M \rightarrow Y$ (typically on the standardized scale).
var_a, var_b	Variances (squared standard errors) of $\hat{a}$ and $\hat{b}$. For standardized regression with no covariates, $\mathrm{Var}(\hat{a}) \approx (1-a^2)/(n-2)$ and $\mathrm{Var}(\hat{b}) \approx (1-b^2)/\{(n-3)(1-a^2)\}$, the standardized-model variance accounting for the correlation the a path induces between the predictors of Y.
cov_ab	Optional covariance between $\hat{a}$ and $\hat{b}$. Defaults to 0 (the assumption underlying the standard Sobel formula). In practice the two estimators are nearly uncorrelated when the controls in the Y-on-M regression are uncorrelated with the M-on-X regression's predictors.

Details

Sobel (1982) first-order. The delta method variance of $\hat{a}\hat{b}$ under independent $\hat{a}$, $\hat{b}$ is

$$\mathrm{Var}_{\mathrm{Sobel}}(\hat{a}\hat{b}) \;=\; a^2\mathrm{Var}(\hat{b}) + b^2\mathrm{Var}(\hat{a}).$$

This is the most cited form and is the variance used by the standard Sobel z-test (Sobel, 1982).

Aroian (1947). Aroian retains the second-order term:

$$\mathrm{Var}_{\mathrm{Aroian}}(\hat{a}\hat{b}) \;=\; a^2\mathrm{Var}(\hat{b}) + b^2\mathrm{Var}(\hat{a}) + \mathrm{Var}(\hat{a})\,\mathrm{Var}(\hat{b}).$$

Aroian shows this is exact under joint normality of the two independent estimators.

Goodman (1960). Goodman's "unbiased" variance subtracts the second-order term instead of adding it:

$$\mathrm{Var}_{\mathrm{Goodman}}(\hat{a}\hat{b}) \;=\; a^2\mathrm{Var}(\hat{b}) + b^2\mathrm{Var}(\hat{a}) - \mathrm{Var}(\hat{a})\,\mathrm{Var}(\hat{b}).$$

For small variances the three forms agree to leading order; they diverge for noisy $\hat{a}$, $\hat{b}$.

Second-order delta method (with covariance). When $\hat{a}$ and $\hat{b}$ share variability (e.g., they are both estimated from the same regression of Y on X and M), the cross-product covariance term enters:

$$\mathrm{Var}(\hat{a}\hat{b}) \;\approx\; a^2\mathrm{Var}(\hat{b}) + b^2\mathrm{Var}(\hat{a}) + 2ab\cdot\mathrm{Cov}(\hat{a},\hat{b}).$$

MacKinnon et al.\ (2002) show this matters in models with covariates that simultaneously load on M and Y.

Connection to `ss_aipe_indirect_effect`. The Sobel (delta method) variance is what `ss_aipe_indirect_effect` builds on under `method = "closed_form"` for AIPE planning; this function makes the alternative formulas available for explicit comparison.

Value

A `data.frame` with rows for the four variance formulas; columns are `term` and `value`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Aroian, L. A. (1947). The probability function of the product of two normally distributed variables. *The Annals of Mathematical Statistics, 18*(2), 265–271.

Goodman, L. A. (1960). On the exact variance of products. *Journal of the American Statistical Association, 55*(292), 708–713.

Lachowicz, M. J., Preacher, K. J., & Kelley, K. (2018). A novel measure of effect size for mediation analysis. *Psychological Methods, 23*, 244–261. doi:10.1037/met0000165

MacKinnon, D. P., Lockwood, C. M., Hoffman, J. M., West, S. G., & Sheets, V. (2002). A comparison of methods to test mediation and other intervening variable effects. *Psychological Methods, 7*(1), 83–104. doi:10.1037/1082989X.7.1.83

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Preacher, K. J., & Kelley, K. (2011). Effect size measures for mediation models: Quantitative strategies for communicating indirect effects. *Psychological Methods, 16*(2), 93–115. doi:10.1037/a0022658

Sobel, M. E. (1982). Asymptotic confidence intervals for indirect effects in structural equation models. *Sociological Methodology, 13*, 290–312.

Tofighi, D., & Kelley, K. (2020). Improved inference in mediation analysis: Introducing the model-based constrained optimization procedure. *Psychological Methods, 25*, 496–515. doi:10.1037/met0000259

See Also

`ss_aipe_indirect_effect`

Other variance utilities: `var_alpha()`, `var_cv()`, `var_ete()`, `var_omega_squared()`, `var_r()`, `var_smd()`, `var_smd_trimmed()`

Examples

```
# 1. a = 0.40, b = 0.40, var_a = 0.02, var_b = 0.02, no covariance:
var_indirect_effect(a = 0.40, b = 0.40, var_a = 0.02, var_b = 0.02)

# 2. With a positive covariance between a-hat and b-hat:
var_indirect_effect(a = 0.40, b = 0.40,
                    var_a = 0.02, var_b = 0.02, cov_ab = 0.005)
```

`var_omega_squared` *Asymptotic Variance of Omega Squared (ANOVA Effect Size)*

Description

Computes the large-sample (delta method) variance of the sample $\hat{\omega}^2$ (Hays' 1994 bias-corrected estimator) in a fixed-effects ANOVA. Fleishman (1980, Eq. 22, p. 669) gives the exact variance of the unbiased estimator of the signal-to-noise ratio $f^2 = \sigma_a^2/\sigma_e^2$ under the noncentral F sampling distribution of the observed F statistic; because $\omega^2 = f^2/(1+f^2)$ (his Eq. 8), the delta method carries that variance to the ω^2 scale with the Jacobian $\mathrm{d}\omega^2/\mathrm{d}f^2 = (1-\omega^2)^2$. Fleishman gives no variance on the ω^2 scale himself, so the transfer is this package's step rather than his. The result is the natural companion to `ci_omega_squared` (CI) and `omega_squared` (point estimate).

Usage

```
var_omega_squared(
  population_omega_squared = NULL,
  df_effect = NULL,
  df_error = NULL,
  N = NULL,
  object = NULL
)
```

Arguments

`population_omega_squared`	Population ω^2. Numeric scalar in $[0, 1)$. Ignored when `object` is supplied.
`df_effect`	Numerator degrees of freedom for the effect. Ignored when `object` is supplied.
`df_error`	Error (residual) degrees of freedom. Ignored when `object` is supplied.
`N`	Total sample size. Ignored when `object` is supplied.
`object`	Optional fitted `aov` or `lm` object. When supplied, the function loops over the non-`Residuals` rows of `anova(object)` and returns one row per effect, plugging in the sample $\hat{\omega}_p^2$ for each as the working population value.

Details

Derivation. In a fixed-effects ANOVA with numerator df df_1 and denominator df df_2, the observed F statistic follows a noncentral F distribution with noncentrality $\lambda = df_1(F-1)$ when $\hat{\omega}^2 = df_1(F-1)/[df_1(F-1)+N]$ is the population value (Hays, 1994). The asymptotic variance of $\hat{\omega}^2$ is obtained by the delta method on this relationship (Fleishman, 1980), yielding:

$$\mathrm{Var}(\hat{\omega}^2) \;\approx\; \frac{2 \cdot df_1 \cdot (df_2 - 2)(1-\omega^2)^2(1+\lambda^*/df_1)^2}{N^2(df_2-4)},$$

with $\lambda^* = \omega^2 N/(1-\omega^2)$ the noncentrality implied by the population value. This is the form used by `ci_omega_squared` when constructing a Wald-style interval; the noncentral F CI is generally preferred.

Caveats. The variance is a delta method approximation; it becomes inaccurate when df_2 is small (< 10), when ω^2 is near the boundaries 0 or 1, or when the residual distribution is heavy-tailed. For small-sample inference, the noncentral F CI (`ci_omega_squared`) is preferred over a Wald-style interval built on this variance.

Value

A 1-row `data.frame` with columns `term` (`"var_omega_squared"`) and `value` (the asymptotic variance).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Fleishman, A. I. (1980). Confidence intervals for correlation ratios. *Educational and Psychological Measurement, 40*(3), 659–670.

Hays, W. L. (1994). *Statistics* (5th ed.). Fort Worth, TX: Harcourt Brace College Publishers.

Kelley, K. (2007). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20*(8), 1–24. doi:10.18637/jss.v020.i08

Kelley, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*, 137–152. doi:10.1037/a0028086

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on η^2, Chapter 7 on factorial designs, and Chapter 11 on generalized η^2 for within-subjects designs.)

Olejnik, S., & Algina, J. (2003). Generalized eta and omega squared statistics: Measures of effect size for some common research designs. *Psychological Methods, 8*(4), 434–447. doi:10.1037/1082989X.8.4.434

Steiger, J. H. (2004). Beyond the *F* test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods, 9*(2), 164–182. doi:10.1037/1082989X.9.2.164

See Also

`omega_squared`, `omega_squared_partial`, `ci_omega_squared`, `ss_aipe_omega_squared`

Other variance utilities: `var_alpha()`, `var_cv()`, `var_ete()`, `var_indirect_effect()`, `var_r()`, `var_smd()`, `var_smd_trimmed()`

Examples

```
# 1. One way ANOVA: 3 groups (df_effect = 2), 60 total (df_error = 57),
#        population omega^2 = 0.10.
var_omega_squared(population_omega_squared = 0.10,
                  df_effect = 2,
                  df_error = 57,
                  N = 60)

# 2. Per-effect variance from a fitted lm() / aov() (pygmalion data:
#        expectancy treatment x grade, N = 310):
fit_factorial <- aov(iq_8 ~ treatment * factor(grade), data = pygmalion)
var_omega_squared(object = fit_factorial)
```

`var_partial_r` *Asymptotic Variance of the Partial Correlation Coefficient*

Description

Computes the large-sample variance of the sample partial correlation coefficient $r_{XY \cdot Z_1 \cdots Z_J}$ under multivariate normality. Two formulas are available: the classical asymptotic variance on the raw r scale (the default), and the variance $1/(n - J - 3)$ of the Fisher's *Z* transformation (Fisher, 1921, 1924), which is the appropriate quantity for constructing a confidence interval by transformation and back-transformation.

Usage

```
var_partial_r(r, n, J = 1, fisher_z = FALSE)
```

Arguments

`r`	The sample partial correlation coefficient $r_{XY \cdot Z_1 \cdots Z_J}$. Must be in $[-1, 1]$.
`n`	Total sample size.
`J`	Number of variables partialled out (i.e., the count of $Z_1, \ldots, Z_J$); must be at least 1. Defaults to 1.
`fisher_z`	Logical. If `FALSE` (the default), the function returns the raw-scale asymptotic variance. If `TRUE`, it returns the variance of the Fisher's *Z* transformation $Z = \operatorname{arctanh}(r)$ under the Fisher (1924) reduction.

Details

Raw-scale asymptotic variance. Under multivariate normality the partial correlation $\hat{r}_{XY \cdot Z}$ has the large-sample variance (Fisher, 1924, for the reduction; the simple-correlation building block is, e.g., Olkin & Finn, 1995, their Equation 3):

$$\operatorname{Var}(\hat{r}_{XY \cdot Z}) \;\approx\; \frac{(1 - \rho^2_{XY \cdot Z})^2}{n - J - 1},$$

a direct generalization of the classical asymptotic variance $(1 - \rho^2)^2/(n - 1)$ of the simple Pearson correlation (Fisher, 1915) with the degrees of freedom reduced by the number of partialled variables. The function evaluates this with $\hat{r}$ substituted for ρ.

Fisher's *Z* Transformation. Fisher (1921) showed that for a Pearson correlation, the transformation $Z = \frac{1}{2}\log\{(1 + r)/(1 - r)\} = \operatorname{arctanh}(r)$ is approximately normal with variance $1/(n - 3)$. Fisher (1924) showed that the partial correlation based on n observations with J variables partialled out is distributed as a simple correlation from a sample reduced in size by J; combined with the Fisher (1921) variance of Z, the Fisher's *Z* transformation of $\hat{r}_{XY \cdot Z_1 \cdots Z_J}$ is therefore approximately normal with variance $1/(n - J - 3)$. This is the standard ingredient for constructing a confidence interval on $\rho_{XY \cdot Z}$ by transforming, building a Wald interval on Z, and back-transforming with $\tanh$.

Value

A one-row `data.frame` with columns `term` (either `"var_partial_r"` or `"var_fisher_z_partial_r"`) and `value` (the requested variance).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (3rd ed.). Lawrence Erlbaum.

Fisher, R. A. (1915). Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika, 10*(4), 507–521.

Fisher, R. A. (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron, 1*, 3–32.

Fisher, R. A. (1924). The distribution of the partial correlation coefficient. *Metron, 3*, 329–332.

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on contrasts, Chapter 5 on multiple comparisons, and Chapter 9 on ANCOVA.)

Olkin, I., & Finn, J. D. (1995). Correlations redux. *Psychological Bulletin, 118*(1), 155–164. doi:10.1037/00332909.118.1.155

See Also

`var_semipartial_r`, `var_R2`, `convert_r_Z`, `ci_r`

Examples

```
# Olkin-Finn (1995) asymptotic variance for r_p = .35, n = 80,
#     J = 2 control variables.
var_partial_r(r = 0.35, n = 80, J = 2)

# Variance of Fisher's Z transformation (Fisher, 1921, 1924) for the
#     same setting, useful for building a CI on rho_p.
var_partial_r(r = 0.35, n = 80, J = 2, fisher_z = TRUE)
```

var_r *Asymptotic Variance of the Pearson Correlation Coefficient*

Description

Computes the asymptotic (large-sample) variance of the sample Pearson product-moment correlation r under bivariate normality (Fisher, 1915) and, optionally, the Bonett-Wright (2000) kurtosis-corrected variance for non-normal margins. Stand-alone variance utility: surprisingly absent from CRAN despite being a building block for AIPE planning, meta-analytic weighting, and Wald-style inference on r.

Usage

```
var_r(rho, n, kurtosis_x = NULL, kurtosis_y = NULL)
```

Arguments

rho	Population correlation coefficient. Numeric scalar or vector in $(-1, 1)$.
n	Total sample size on which the Pearson r would be computed. Scalar or vector.
kurtosis_x	Optional excess kurtosis of the marginal distribution of X. When kurtosis_x and kurtosis_y are both supplied (along with rho_xy_2x2y if available), the Bonett-Wright (2000) corrected variance is returned in addition to the normal-theory variance.
kurtosis_y	Optional excess kurtosis of Y; see kurtosis_x.

Details

Normal-theory variance (default). Under bivariate normality the large-sample variance is

$$\mathrm{Var}(\hat{r}) \;\approx\; (1-\rho^2)^2/(n-1),$$

the leading term of the exact moment expansion (Hotelling, 1953, Section 7; the exact density of r is Fisher's, 1915). This is the workhorse variance and is exact in the limit; it is also what Fisher's Z CI ci_r uses on the transformed scale.

Bonett-Wright kurtosis correction. When the marginals are *not* normal, the asymptotic variance picks up a kurtosis- dependent correction (Bonett & Wright, 2000):

$$\mathrm{Var}(\hat{r}) \;\approx\; (1-\rho^2)^2/(n-1) \cdot \big(1 + \rho^2(\gamma_2^{(X)} + \gamma_2^{(Y)})/4\big),$$

where $\gamma_2^{(X)}$, $\gamma_2^{(Y)}$ are the excess kurtoses of the two marginals. The correction is exact when the joint distribution is elliptical; for non-elliptical joints it is a first-order approximation. When the kurtosis arguments are NULL, only the normal-theory variance is returned.

Connection to Fisher's Z transform. On the variance- stabilized scale $Z = \tanh^{-1}(r)$ the asymptotic variance is $1/(n-3)$ regardless of ρ (Fisher, 1921). This is reported alongside the raw-scale variance because it is the natural working scale for CI construction (ci_r).

Value

A `data.frame` with the rows

- `var_r_normal`, the normal-theory asymptotic variance $(1 - \rho^2)^2/(n - 1)$ (Hotelling, 1953, Section 7),
- `var_r_bonett_wright` (when kurtoses are supplied) the Bonett & Wright (2000) kurtosis-corrected variance,
- `var_fisher_z`, the variance of the Fisher Z transform, $1/(n - 3)$.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Bonett, D. G., & Wright, T. A. (2000). Sample size requirements for estimating Pearson, Kendall and Spearman correlations. *Psychometrika, 65*(1), 23–28. doi:10.1007/BF02294183

Fisher, R. A. (1915). Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika, 10*(4), 507–521.

Fisher, R. A. (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron, 1*, 3–32.

Hotelling, H. (1953). New light on the correlation coefficient and its transforms. *Journal of the Royal Statistical Society, Series B, 15*(2), 193–232. (Section 7 gives the exact moments of r.)

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge.

Olkin, I., & Finn, J. D. (1995). Correlations redux. *Psychological Bulletin, 118*(1), 155–164. doi:10.1037/00332909.118.1.155

See Also

`expected_r`, `ci_r`, `var_partial_r`, `var_semipartial_r`

Other variance utilities: `var_alpha()`, `var_cv()`, `var_ete()`, `var_indirect_effect()`, `var_omega_squared()`, `var_smd()`, `var_smd_trimmed()`

Examples

```
# 1. Normal-theory variance:
var_r(rho = 0.30, n = 50)

# 2. With Bonett-Wright correction for leptokurtic margins
#        (excess kurtosis = 3 for each variable):
var_r(rho = 0.30, n = 50, kurtosis_x = 3, kurtosis_y = 3)
```

var_R2 *Variance of the Squared Multiple Correlation Coefficient*

Description

Computes the sampling variance of the squared multiple correlation coefficient from the population value, the sample size, and the number of predictors, the quantity that governs how precisely R^2 is estimated at a given design size.

Usage

```
var_R2(population_R2, N, p)
```

Arguments

`population_R2`	Population squared multiple correlation coefficient
`N`	Sample size
`p`	The number of predictor variables

Details

Uses the hypergeometric function as discussed in and section 28 of Stuart, Ord, and Arnold (1999) in order to obtain the *correct* value for the variance of the squared multiple correlation coefficient.

Value

A 1-row `data.frame` with columns `term` and `value`. The `term` value is `"var_R2"` and `value` is the asymptotic variance of R^2.

Note

The Gauss hypergeometric function ${}_2F_1$ is computed in base R (see the internal `.hyperg_2F1`); no GSL system library is required.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Kelley, K. (2008). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals. *Multivariate Behavioral Research, 43*, 524–555. doi:10.1080/00273170802490632

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 3 on R^2 as a model comparison effect size.)

Stuart, A., Ord, J. K., & Arnold, S. (1999). *Kendall's advanced theory of statistics, volume 2A: Classical inference and the linear model* (6th ed.). Arnold.

See Also

expected_R2, ci_R2, ss_aipe_R2

Examples

```
var_R2(.5, 10, 5)
var_R2(.5, 25, 5)
var_R2(.5, 50, 5)
var_R2(.5, 100, 5)
```

---

var_semipartial_r *Asymptotic Variance of the Semipartial (Part) Correlation Coefficient*

Description

Computes the large-sample variance of the sample semipartial (also known as the part) correlation coefficient $r_{Y(X \cdot Z_1 \cdots Z_J)}$ under multivariate normality. The function provides two variances: the asymptotic variance under the alternative (the partial-correlation form applied by analogy) and, when the full-model coefficient of multiple determination $R^2_{Y \cdot XZ_1 \cdots Z_J}$ is supplied, the exact null-hypothesis variance derived from the multiple-regression F-test for the unique contribution of X (Cohen, Cohen, West, & Aiken, 2003).

Usage

```
var_semipartial_r(r_sp, n, J = 1, R2_full = NULL)
```

Arguments

r_sp	Sample semipartial correlation coefficient $r_{Y(X \cdot Z_1 \cdots Z_J)}$, with the controlled variables partialled out of X only (not Y). Must be in $[-1, 1]$.
n	Total sample size.
J	Number of variables partialled out of X (i.e., the count of $Z_1, \ldots, Z_J$); must be at least 1. Defaults to 1.
R2_full	Optional coefficient of multiple determination $R^2_{Y \cdot XZ_1 \cdots Z_J}$ from the full model that includes X and all controls. When supplied, the null-hypothesis variance $(1 - R^2)/(n - J - 2)$ is returned instead of the alternative-side asymptotic variance. Must be in $[0, 1]$.

Details

Background. The squared semipartial $r^2_{Y(X \cdot Z)}$ equals the increase in R^2 when X is added to a model already containing the controls $Z_1, \ldots, Z_J$, i.e., the unique variance in Y attributable to X. Unlike the partial, the semipartial is on the original scale of Y rather than on the partialled scale, which makes it the natural effect size companion to standardized regression coefficients in multiple-regression reports (Cohen et al., 2003).

Asymptotic variance (default). Under multivariate normality the semipartial admits the same large-sample form as the partial (Fisher, 1924, applied by analogy):

$$\mathrm{Var}(\hat{r}_{Y(X\cdot Z)}) \;\approx\; \frac{(1-\rho^2_{Y(X\cdot Z)})^2}{n-J-1}.$$

The function evaluates this with $\hat{r}_{sp}$ substituted for ρ_{sp}. This is the appropriate quantity for Wald-style inference and for AIPE-style precision planning analogous to that of the partial correlation. Aloe and Becker (2012) develop the asymptotic variance of the semipartial as a function of the full population correlation structure, and Yuan and Chan (2011) give exact higher-order results for the closely related standardized regression coefficients; the present approximation matches the leading $1/n$ behavior.

Null-hypothesis variance (when `R2_full` is supplied). In multiple regression the unique contribution of X is tested with

$$F \;=\; \frac{r^2_{Y(X\cdot Z)}\,(n-J-2)}{1-R^2_{Y\cdot XZ}} \;\sim\; F(1,\,n-J-2)$$

under $H_0\colon \rho_{Y(X\cdot Z)}=0$ (Cohen et al., 2003, equation 3.7.3). Equivalently $t=\hat{r}_{sp}\sqrt{(n-J-2)/(1-R^2_{Y\cdot XZ})}$ is a t-statistic on $n-J-2$ degrees of freedom, so the *under-the-null* variance of $\hat{r}_{sp}$ is

$$\mathrm{Var}_0(\hat{r}_{sp}) \;=\; \frac{1-R^2_{Y\cdot XZ}}{n-J-2}.$$

Supplying `R2_full` returns this null variance, which is the standard ingredient for testing the significance of X's unique contribution.

Value

A one-row `data.frame` with columns `term` (either `"var_semipartial_r"` or `"var_semipartial_r_under_null"`) and `value` (the requested variance).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Aloe, A. M., & Becker, B. J. (2012). An effect size for regression predictors in meta-analysis. *Journal of Educational and Behavioral Statistics, 37*(2), 278–297. doi:10.3102/1076998610396901

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (3rd ed.). Lawrence Erlbaum.

Fisher, R. A. (1924). The distribution of the partial correlation coefficient. *Metron, 3*, 329–332.

Kelley, K., & Maxwell, S. E. (2003). Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*(3), 305–321. doi:10.1037/1082989X.8.3.305

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on contrasts, Chapter 5 on multiple comparisons, and Chapter 9 on ANCOVA.)

Yuan, K.-H., & Chan, W. (2011). Biases and standard errors of standardized regression coefficients. *Psychometrika, 76*(4), 670–690. doi:10.1007/s1133601192246

See Also

var_partial_r, var_R2, ci_reg_coef

Examples

```
# Olkin-Finn-style asymptotic variance of a semipartial r = .25 with
#     n = 100 and J = 3 controls in X.
var_semipartial_r(r_sp = 0.25, n = 100, J = 3)

# With R^2 of the full model also supplied, the function returns the
#     null-hypothesis variance used in the F-test for X's unique
#     contribution.
var_semipartial_r(r_sp = 0.25, n = 100, J = 3, R2_full = 0.42)
```

---

var_smd *Variance of Cohen's* d *and Hedges'* g

Description

Computes the variance of the sample standardized mean difference (Cohen's d) under bivariate normality and homogeneous variances, using the *exact* noncentral t sampling distribution (Hedges, 1981) as the default and reporting the large-sample Hedges-Olkin (1985) approximation alongside for comparison. Optionally returns the variance of Hedges' g, the bias-corrected counterpart of d.

Usage

```
var_smd(delta, n_1, n_2 = NULL, unbiased = FALSE)
```

Arguments

delta	Population standardized mean difference. Numeric scalar or vector.
n_1	Sample size in group 1. Scalar or vector.
n_2	Sample size in group 2. Scalar or vector. Defaults to n_1 (balanced design).
unbiased	Logical. If TRUE, returns the variance of Hedges' g (the bias-corrected estimator); if FALSE (the default), returns the variance of Cohen's d.

Details

var_smd() is a stand-alone variance utility: most R packages return only the Hedges-Olkin large-sample approximation and conflate the variances of d and g (Goulet-Pelletier & Cousineau, 2018). The drift between the exact and approximate forms becomes non-trivial below about $n = 30$ per group and matters whenever var_smd() feeds into meta-analytic weighting, AIPE planning, or a Wald-style standard-error report.

Exact noncentral *t* form. For $\hat{d} = (\bar{Y}_1 - \bar{Y}_2)/s_p$ with pooled s_p, the rescaled statistic $t = \hat{d}\sqrt{n_1 n_2/(n_1+n_2)}$ follows a noncentral t with $df = n_1 + n_2 - 2$ degrees of freedom and noncentrality parameter $\lambda = \delta\sqrt{n_1 n_2/(n_1+n_2)}$. The variance of a noncentral t is (Johnson, Kotz, & Balakrishnan, 1995, Sec.\ 31.3)

$$\mathrm{Var}(t) \;=\; \frac{df\,(1+\lambda^2)}{df-2} \;-\; \lambda^2\,c(df)^2,$$

where $c(df) = \sqrt{df/2}\,\Gamma((df-1)/2)\,/\,\Gamma(df/2)$; dividing by the design factor $n_1 n_2/(n_1+n_2)$ returns $\mathrm{Var}(\hat{d})$. For Hedges' g, multiply the result by $J(df)^2$ where $J(df) = 1/c(df)$ is the Hedges-Olkin (1985) bias-correction factor (see `expected_smd`).

Hedges-Olkin large-sample approximation. The frequently quoted approximation (Hedges & Olkin, 1985, equation 8) is

$$\mathrm{Var}(\hat{d}) \;\approx\; \frac{n_1+n_2}{n_1 n_2} \;+\; \frac{\delta^2}{2(n_1+n_2-2)}.$$

This approaches the exact form only as the degrees of freedom grow: even at $\delta = 0$ it returns $1/(n_1 n_2/(n_1+n_2))$ while the exact noncentral t variance is $[df/(df-2)]/(n_1 n_2/(n_1+n_2))$, so the approximation is biased downward by a factor of $(df-2)/df$, and the downward bias grows with δ and small n. Goulet-Pelletier & Cousineau (2018) document the drift and recommend the exact form for $n < 30$ per group.

Companions. `var_smd()` is the variance partner of `expected_smd` (mean) and `ci_smd` (CI). For design-stage AIPE planning that solves for n given a target CI width on d, see `ss_aipe_smd`.

Value

A `data.frame` with rows for the exact (noncentral- t) variance and the Hedges-Olkin large-sample approximation. Columns are `term` (`"var_smd_exact"` or `"var_smd_approx"`) and `value`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Goulet-Pelletier, J.-C., & Cousineau, D. (2018). A review of effect sizes and their confidence intervals, Part I: The Cohen's *d* family. *The Quantitative Methods for Psychology, 14*(4), 242–265. doi:10.20982/tqmp.14.4.p242

Hedges, L. V. (1981). Distribution theory for Glass's estimator of effect size and related estimators. *Journal of Educational Statistics, 6*(2), 107–128.

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. Academic Press.

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous univariate distributions, volume 2* (2nd ed.). Wiley.

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons and Chapter 3 on one-way ANOVA.)

See Also

smd, ci_smd, expected_smd, ss_aipe_smd

Other variance utilities: var_alpha(), var_cv(), var_ete(), var_indirect_effect(), var_omega_squared(), var_r(), var_smd_trimmed()

Examples

```
# 1. Balanced design, delta = 0.5, n = 20 per group.
var_smd(delta = 0.5, n_1 = 20)

# 2. Hedges-Olkin approximation drifts from exact when n is small.
var_smd(delta = 0.5, n_1 = 5)
var_smd(delta = 0.5, n_1 = 50)

# 3. Variance of Hedges' g (bias-corrected).
var_smd(delta = 0.5, n_1 = 20, unbiased = TRUE)
```

---

var_smd_trimmed *Asymptotic Variance of the Robust Trimmed SMD*

Description

Computes the asymptotic variance of the Algina-Keselman-Penfield (2005) robust standardized mean difference under the Yuen (1974) trimmed-mean framework, suitable for AIPE sample size planning for robust effect sizes (Keselman, Algina, Lix, Wilcox, & Deering, 2008).

Usage

```
var_smd_trimmed(population_smd_trimmed, n_1, n_2, trim = 0.2)
```

Arguments

population_smd_trimmed	Anticipated population value of the robust trimmed SMD δ_R.
n_1, n_2	Per-group sample sizes.
trim	Proportion to trim and Winsorize. Default 0.20.

Details

Variance formula. Under random sampling with trimming proportion γ from each tail, the variance of the trimmed-mean difference scales by $1/h_j$ (where $h_j = n_j - 2\lfloor \gamma n_j \rfloor$ is the number of retained observations in group j) rather than $1/n_j$. The large-sample variance of the standardized version, written on the d_R scale, is

$$\mathrm{Var}(\hat{d}_R) \;\approx\; \frac{h_1 + h_2}{h_1 h_2} + \frac{\delta_R^2}{2(h_1 + h_2)}.$$

For $\gamma = 0$ this reduces to the standard Hedges-Olkin (1985) variance of Cohen's *d*.

When to use. For AIPE planning of a robust effect size study, use `var_smd_trimmed()` in place of `var_smd()`. Pair with `smd_trimmed()` for the point estimate and noncentral *t* CI.

Value

A 1-row `data.frame` with columns `term` (`"var_smd_trimmed"`) and `value` (the variance).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Algina, J., Keselman, H. J., & Penfield, R. D. (2005). An alternative to Cohen's standardized mean difference effect size: A robust parameter and confidence interval in the two independent groups case. *Psychological Methods, 10*(3), 317–328. doi:10.1037/1082989X.10.3.317

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in parameter estimation via narrow confidence intervals. *Psychological Methods, 11*(4), 363–385. doi:10.1037/1082989X.11.4.363

Keselman, H. J., Algina, J., Lix, L. M., Wilcox, R. R., & Deering, K. N. (2008). A generally robust approach for testing hypotheses and setting confidence intervals for effect sizes. *Psychological Methods, 13*(2), 110–129. doi:10.1037/1082989X.13.2.110

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2027). *Designing experiments and analyzing data: A model comparison perspective* (4th ed.). Routledge. (See Chapter 4 on individual comparisons and Chapter 3 on one-way ANOVA.)

Yuen, K. K. (1974). The two-sample trimmed *t* for unequal population variances. *Biometrika, 61*(1), 165–170.

See Also

`smd_trimmed`, `var_smd`, `ss_aipe_smd`

Other variance utilities: `var_alpha()`, `var_cv()`, `var_ete()`, `var_indirect_effect()`, `var_omega_squared()`, `var_r()`, `var_smd()`

Examples

```
# 1. Population delta_R = 0.5, n = 30 per group, 20% trim:
var_smd_trimmed(population_smd_trimmed = 0.5, n_1 = 30, n_2 = 30)

# 2. Variance scales by h / n via the trimming proportion:
var_smd_trimmed(0.5, 30, 30, trim = 0.00)$value
var_smd_trimmed(0.5, 30, 30, trim = 0.20)$value
```

---

vargha_delaney_A *Vargha and Delaney's* A *(Stochastic-Superiority Effect Size)*

Description

Computes Vargha and Delaney's (2000) *A*, the probability that a randomly drawn observation from the first sample exceeds a randomly drawn observation from the second (with tied pairs counted as half), together with its asymptotic standard error from DeLong, DeLong, and Clarke-Pearson (1988) and a confidence interval on the population *A*. *A* is equivalent to the receiver-operating-characteristic area-under-the-curve (AUC) and to the common-language effect size (McGraw & Wong, 1992) under a continuous-response assumption, and is a robust, scale-free ordinal effect size that does not require equal variances or normality.

Usage

```
vargha_delaney_A(
  x,
  y = NULL,
  data = NULL,
  conf_level = 0.95,
  ci_method = c("logit", "wald")
)
```

Arguments

x	Either a numeric vector of observations from group 1, or a two-sided formula of the form `outcome ~ group` (in which case `data` must be supplied and the grouping variable must have exactly two levels).
y	Numeric vector of observations from group 2. Ignored when `x` is a formula.
data	Optional data frame containing the variables named in the formula `x`.
conf_level	Confidence coverage for a symmetric interval (default `0.95`).
ci_method	Either `"logit"` (default; Wald interval on the logit of *A* with back-transformation, recommended for finite samples; Newcombe, 2006b) or `"wald"` (untransformed Wald on the original scale, clipped to [0, 1]).

Details

Definition. For independent samples $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$,

$$A = \Pr(X > Y) + \tfrac{1}{2} \Pr(X = Y).$$

Values of $A = 0.5$ indicate stochastic equality; $A > 0.5$ indicates that group 1 tends to score higher. Qualitative magnitude labels for *A* are not reported here, in keeping with the DMAR convention of reporting effect sizes as numbers with confidence intervals.

Sample estimate. Equivalent rank-based computation:

$$\hat{A} = \frac{\bar{R}_X - (n_1 + 1)/2}{n_2},$$

where $\bar{R}_X$ is the mean rank of the first sample in the pooled ranking with mid-ranks for ties (Vargha & Delaney, 2000, p. 109). Equivalently, $\hat{A} = U/(n_1 n_2)$, with U the Mann-Whitney U-statistic counting $X_i > Y_j$ (tied pairs at $1/2$).

Standard error. The function uses the DeLong-DeLong-Clarke-Pearson (1988) U-statistic variance estimator, which is unbiased under sampling from any joint distribution (no parametric or homoscedasticity assumption). Defining the placement components

$$V_{10}(X_i) = \frac{1}{n_2} \sum_j \psi(X_i, Y_j), \qquad V_{01}(Y_j) = \frac{1}{n_1} \sum_i \psi(X_i, Y_j),$$

with $\psi(x, y) = 1, \frac{1}{2}, 0$ as $x > y, =, <$, the variance estimate is

$$\widehat{\mathrm{Var}}(\hat{A}) = \frac{S_{10}^2}{n_1} + \frac{S_{01}^2}{n_2},$$

where S_{10}^2 and S_{01}^2 are the sample variances of the V_{10} and V_{01} placement components. This is identical to the (single-curve) DeLong AUC variance and is the standard nonparametric variance for the Mann-Whitney functional (Brunner & Munzel, 2000).

Confidence interval. `ci_method = "logit"` (the default) constructs a Wald interval on $\mathrm{logit}(A) = \log\{A/(1-A)\}$ using the delta method standard error $\widehat{\mathrm{SE}}(\hat{A})/\{\hat{A}(1-\hat{A})\}$ and back- transforms with the inverse logit. Newcombe (2006a, 2006b) showed in extensive coverage simulations that logit-Wald has notably better small- sample coverage than untransformed Wald, while remaining simple and free of iteration. `ci_method = "wald"` returns the untransformed Wald interval, clipped to $[0, 1]$.

Value

A one-row `data.frame` with columns `A` (point estimate), `se` (DeLong-DeLong-Clarke-Pearson standard error), `lower_limit` and `upper_limit` (confidence limits at `conf_level`), `z_value` and `p_value` (Wald test of $H_0 : A = 0.5$, i.e., stochastic equality), `n_1` and `n_2` (group sample sizes), and `ci_method`.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Brunner, E., & Munzel, U. (2000). The nonparametric Behrens-Fisher problem: Asymptotic theory and a small-sample approximation. *Biometrical Journal, 42*(1), 17–25. doi:10.1002/(SICI)1521-4036(200001)42:1<17::AIDBIMJ17>3.0.CO;2U

DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics, 44*(3), 837–845.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 143*(1), 29–36.

McGraw, K. O., & Wong, S. P. (1992). A common language effect size statistic. *Psychological Bulletin, 111*(2), 361–365. doi:10.1037/00332909.111.2.361

Newcombe, R. G. (2006a). Confidence intervals for an effect size measure based on the Mann-Whitney statistic. Part 1: General issues and tail-area-based methods. *Statistics in Medicine, 25*(4), 543–557. doi:10.1002/sim.2323

Newcombe, R. G. (2006b). Confidence intervals for an effect size measure based on the Mann-Whitney statistic. Part 2: Asymptotic methods and evaluation. *Statistics in Medicine, 25*(4), 559–573. doi:10.1002/sim.2324

Vargha, A., & Delaney, H. D. (2000). A critique and improvement of the CL common language effect size statistics of McGraw and Wong. *Journal of Educational and Behavioral Statistics, 25*(2), 101–132. doi:10.3102/10769986025002101

See Also

smd, ci_smd, cohen_kappa

Examples

```
# Two numeric vectors.
set.seed(113)
x <- rnorm(40, mean = 0.6)
y <- rnorm(40, mean = 0)
vargha_delaney_A(x, y)

# Formula interface on the pygmalion field experiment. The first factor
# level (Control) forms group 1, so A below 0.5 says a randomly drawn
# control child tends to score below a child from the bloomer group.
vargha_delaney_A(iq_8 ~ treatment, data = pygmalion)

# Wald (untransformed) interval rather than logit.
vargha_delaney_A(x, y, ci_method = "wald")
```

`variance_components_mls` *Modified-Large-Sample Confidence Intervals on Variance Components*

Description

Computes modified-large-sample (MLS) confidence intervals on the between-group and within-group variance components of a balanced one-way random-effects ANOVA, following Burdick & Graybill (1992). MLS intervals have substantially better coverage than the Satterthwaite or simple-Wald intervals when the components are far from zero, and they are the standard interval method in generalizability theory (Brennan, 2001).

Usage

```
variance_components_mls(
  ms_between,
  ms_within,
  df_between,
  df_within,
  n,
  conf_level = 0.95
)
```

Arguments

`ms_between`	Mean square between groups (numerator of the ANOVA F).
`ms_within`	Mean square within groups (denominator of the ANOVA F).
`df_between`	Degrees of freedom for the between-group MS (typically $a - 1$ for a groups).
`df_within`	Degrees of freedom for the within-group MS (typically $a(n-1)$ for a groups of size n).
`n`	Number of observations per group (assumed balanced).
`conf_level`	Confidence level for the CIs. Default `0.95`.

Details

Point estimates. For a one-way random-effects ANOVA on a groups of size n, the method-of-moments estimators are

$$\hat{\sigma}_b^2 \;=\; \max(0,\, (MS_b - MS_w)/n), \qquad \hat{\sigma}_w^2 \;=\; MS_w.$$

Modified-large-sample CIs (Burdick-Graybill 1992). The MLS interval for σ_b^2 is

$$\left[\frac{MS_b - MS_w - \sqrt{V_L}}{n},\;\; \frac{MS_b - MS_w + \sqrt{V_U}}{n}\right],$$

with

$$V_L \;=\; G_1^2 MS_b^2 + H_2^2 MS_w^2 + G_{12} MS_b MS_w, \qquad V_U \;=\; H_1^2 MS_b^2 + G_2^2 MS_w^2 + H_{12} MS_b MS_w,$$

where the constants G_1, G_2, H_1, H_2 and the cross-term constants G_{12}, H_{12} depend on the degrees of freedom and on χ^2 and F quantiles at the chosen confidence level (Burdick & Graybill, 1992, equations 2.4.1–2.4.5 give the explicit formulas). The lower limit is truncated at zero. For the within-group component, the standard χ^2-based CI on MS_w (Searle, Casella, & McCulloch, 1992) is used.

Caveats. MLS intervals assume balanced data and homogeneous variances within groups. For unbalanced data the appropriate analog is the Burdick-Graybill MLS extension to unequal sample sizes (Burdick & Graybill, 1992, Section 2.5), which is not implemented here.

Value

A `data.frame` with rows for the point estimates and MLS lower / upper CIs of the between-group variance component (σ_b^2), the within-group component (σ_w^2), and the implied intraclass correlation ($\rho = \sigma_b^2/(\sigma_b^2 + \sigma_w^2)$).

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Brennan, R. L. (2001). *Generalizability theory*. Springer.

Burdick, R. K., & Graybill, F. A. (1992). *Confidence intervals on variance components*. Marcel Dekker.

Searle, S. R., Casella, G., & McCulloch, C. E. (1992). *Variance components*. Wiley.

See Also

`icc`, `var_icc`, `ss_aipe_icc`

Other agreement and measurement: `R2_mixed_effects()`, `content_validity_index()`, `gwet_ac()`, `icc_lmer()`, `krippendorff_alpha()`, `limits_of_agreement()`, `lin_ccc()`

Examples

```
# 1. Balanced one-way random-effects ANOVA: a = 10 groups, n = 5.
#        Hypothetical MS_b = 6.0, MS_w = 1.5.
variance_components_mls(ms_between = 6.0, ms_within = 1.5,
                        df_between = 9, df_within = 40, n = 5)
```

`welch_t` *Welch's Separate-Variance* t *Test*

Description

Computes Welch's (1947) separate-variance *t* test, the Satterthwaite (1946) approximation for unequal-variance two-sample inference, and returns the test statistic, Satterthwaite degrees of freedom, *p*-value, point estimate of the mean difference, and a confidence interval on the mean difference, all in a tidy `data.frame`. Unlike Student's pooled-variance *t* test (`stats::t.test(..., var.equal = TRUE)`), Welch's test does *not* assume the two populations have equal variances, and should be the default choice in applied work (Delacre, Lakens, & Leys, 2017).

Usage

```
welch_t(
  x,
  y,
  mu = 0,
  alternative = c("two_sided", "less", "greater"),
  conf_level = 0.95
)
```

Arguments

`x, y`	Numeric vectors of observations from the two groups. The two groups are independent and need not be the same length; `NA`s are removed from each vector separately.
`mu`	Null value of the mean difference $\mu_1 - \mu_2$. Default `0`.
`alternative`	One of `"two_sided"` (default; the base-R spelling `"two.sided"` is accepted as an alias), `"less"`, or `"greater"`, defining the direction of the alternative hypothesis.
`conf_level`	Confidence level for the CI on the mean difference. Default `0.95`.

Details

Test statistic. Welch's *t* is

$$t \;=\; \frac{\bar{x} - \bar{y} - \mu_0}{\sqrt{s_1^2/n_1 + s_2^2/n_2}},$$

which is referred to a *t* distribution on the Satterthwaite (1946) approximate degrees of freedom

$$df \;=\; \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1 - 1) + (s_2^2/n_2)^2/(n_2 - 1)}.$$

Why Welch by default. Student's pooled-variance *t* assumes $\sigma_1 = \sigma_2$; when that assumption fails it has both inflated and deflated Type I error rates depending on the $n_1 : n_2$ ratio (Ruxton, 2006). Welch's test maintains nominal Type I error across virtually all combinations of σ_1/σ_2 and n_1/n_2,

with no meaningful loss of power when variances are equal. The American Statistical Association and multiple methodological reviews now recommend Welch as the default (Delacre et al., 2017; Lakens, 2015).

Relation to `stats::t.test()`. The numerical results here match `stats::t.test(x, y, var.equal = FALSE)` to machine precision; this function differs only in returning a tidy `data.frame` that composes with the rest of DMAR.

Value

A `data.frame` with rows for the mean difference $\bar{x} - \bar{y}$, the Welch t-statistic, Satterthwaite degrees of freedom, p-value, the CI lower and upper limits on the mean difference, and the per-group means, SDs, and n.

Author(s)

Ken Kelley <kkelley@nd.edu>

References

Delacre, M., Lakens, D., & Leys, C. (2017). Why psychologists should by default use Welch's *t*-test instead of Student's *t*-test. *International Review of Social Psychology, 30*(1), 92–101. doi:10.5334/irsp.82

Lakens, D. (2015, January). Always use Welch's t-test instead of Student's t-test [Blog post]. The 20% Statistician. https://daniellakens.blogspot.com/2015/01/always-use-welchs-t-test-instead-of.html

Ruxton, G. D. (2006). The unequal variance *t*-test is an underused alternative to Student's *t*-test and the Mann-Whitney *U* test. *Behavioral Ecology, 17*(4), 688–690. doi:10.1093/beheco/ark016

Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. *Biometrics Bulletin, 2*(6), 110–114.

Welch, B. L. (1947). The generalization of "Student's" problem when several different population variances are involved. *Biometrika, 34*(1/2), 28–35.

See Also

`t.test`, `summary_t_test`, `smd`, `ci_smd`

Other hypothesis tests: `adjusted_means()`, `ancova()`, `anova_within()`, `ci_dunnett()`, `ci_scheffe()`, `ci_tukey_kramer()`, `compare_cov_structures()`, `contrast_test()`, `correlations_test()`, `equivalence_r()`, `equivalence_smd()`, `factorial_anova()`, `manova_split_plot()`, `mauchly_test()`, `mixed_anova()`, `obrien_test()`, `pairwise_within()`, `randomization_test()`, `randomization_test_paired()`, `regions_of_significance()`, `simple_effects_AB()`, `summary_t_test()`

Examples

```
# 1. Two groups with different variances:
set.seed(113)
x <- rnorm(20, mean = 100, sd = 15)
y <- rnorm(20, mean = 110, sd = 25)
welch_t(x, y)
```

```
# 2. One-sided test:
welch_t(x, y, alternative = "less")

# 3. Side-by-side comparison with base R's stats::t.test().
# The two functions implement the same Welch / Satterthwaite test, so
# the t-statistic, Satterthwaite degrees of freedom, p-value, and the
# CI on the mean difference match exactly. welch_t() differs only in
# what it returns: a data.frame(term, value) rather than a list-
# like htest object. The return composes with dplyr / ggplot2
# pipelines and avoids stringly-typed access like $statistic.
set.seed(113)
a <- rnorm(15, mean = 0,   sd = 1)
b <- rnorm(20, mean = 0.5, sd = 2)

# DMAR (data.frame):
dmar_res <- welch_t(a, b, conf_level = 0.95)
dmar_res

# Base R (htest list):
base_res <- stats::t.test(a, b, var.equal = FALSE, conf.level = 0.95)
base_res

# Verify the four key statistics agree numerically:
pick <- function(term) dmar_res$value[dmar_res$term == term]
stopifnot(
  all.equal(pick("t_statistic"), unname(base_res$statistic)),
  all.equal(pick("df"),          unname(base_res$parameter)),
  all.equal(pick("p_value"),     base_res$p.value),
  all.equal(pick("lower_limit"), base_res$conf.int[1]),
  all.equal(pick("upper_limit"), base_res$conf.int[2])
)
```

`welch_t_broom` *Broom-Style Tidy / Glance Methods for* `welch_t()`

Description

`tidy()` returns the single mean-difference estimate and its confidence interval in the **broom** convention; `glance()` coincides with it, since a two-sample *t* test reports one estimand and there are no extra model-level statistics to add.

Usage

```
## S3 method for class 'dmar_welch_t'
tidy(x, ...)

## S3 method for class 'dmar_welch_t'
glance(x, ...)
```

Arguments

x	A dmar_welch_t object returned by welch_t.
...	Unused.

Value

A one-row data.frame with columns term, estimate, ci_lower, ci_upper, statistic, df, p_value, and conf_level.

Author(s)

Ken Kelley <kkelley@nd.edu>

Index

∗ **AIPE sample size planning**
ss_aipe_c_sensitivity, 524
ss_aipe_cliff_delta, 526
ss_aipe_cliff_delta_sensitivity, 528
ss_aipe_composite_sem, 530
ss_aipe_equivalence_r, 551
ss_aipe_equivalence_r_sensitivity, 553
ss_aipe_equivalence_smd, 556
ss_aipe_equivalence_smd_sensitivity, 558
ss_aipe_icc, 560
ss_aipe_icc_sensitivity, 563
ss_aipe_indirect_effect, 566
ss_aipe_indirect_effect_sensitivity, 571
ss_aipe_mixed_effects_sensitivity, 575
ss_aipe_omega_squared, 577
ss_aipe_omega_squared_sensitivity, 580
ss_aipe_partial_r, 582
ss_aipe_partial_r_sensitivity, 585
ss_aipe_pcm_sensitivity, 590
ss_aipe_r, 592
ss_aipe_r_sensitivity, 594
ss_aipe_reliability_sensitivity, 618
ss_aipe_semipartial_r, 639
ss_aipe_semipartial_r_sensitivity, 641
∗ **Bayesian t analyses**
bayes_independent_t, 29
bayes_one_sample_t, 32
bayes_paired_t, 34
∗ **Cohen's U3**
proportion_of_superiority, 422
∗ **Cohen's d**
ci_smd, 120
smd, 512
∗ **Cronbach's alpha**
reliability_alpha, 445
∗ **Glass's delta**
ci_smd_c, 123
∗ **Hedges' g**
ci_smd, 120
smd, 512
∗ **Vargha-Delaney A**
vargha_delaney_A, 791
∗ **agreement and measurement**
content_validity_index, 153
gwet_ac, 266
icc_lmer, 276
krippendorff_alpha, 287
limits_of_agreement, 291
lin_ccc, 294
R2_mixed_effects, 426
variance_components_mls, 794
∗ **composite power**
ss_power_composite_ancova, 661
ss_power_composite_ancova_2group, 665
ss_power_composite_anova, 672
ss_power_composite_factorial_ancova, 675
ss_power_composite_factorial_ancova_het, 679
ss_power_composite_factorial_anova, 683
ss_power_composite_sem, 686
∗ **confidence intervals for effect sizes**
ci_c, 56
ci_c_ancova, 59
ci_c_ancova_bp, 62
ci_correlation, 64
ci_cv, 68
ci_eta_squared, 73

ci_eta_squared_generalized, 77
ci_eta_squared_partial, 80
ci_mahalanobis, 85
ci_omega_squared, 95
ci_pvaf, 99
ci_R2, 101
ci_rc, 104
ci_reg_coef, 106
ci_rmsea, 109
ci_sc, 111
ci_sc_ancova, 113
ci_sm, 118
ci_smd, 120
ci_smd_c, 123
ci_snr, 125
ci_src, 127
ci_srsnr, 129
contrast_adjusted, 156
plot_smd, 389
* **confidence intervals**
ci_proportion, 97
* **correlation**
expected_partial_r, 247
expected_r, 249
* **critical values**
cv_bonferroni_f, 186
cv_bryant_paulson, 188
cv_chisq, 191
cv_dunnett, 192
cv_f, 194
cv_scheffe, 196
cv_smm, 198
cv_t, 200
cv_tukey_hsd, 201
cv_z, 203
* **data simulators**
simulate_ancova_data, 480
simulate_ancova_factorial_data, 483
simulate_anova_data, 486
simulate_longitudinal_gompertz, 487
simulate_longitudinal_logistic, 491
simulate_longitudinal_negative_exponential, 495
simulate_longitudinal_polynomial, 498
simulate_longitudinal_richards, 504
simulate_regression_data, 508
* **datagen**
simulate_ancova_data, 480
simulate_ancova_factorial_data, 483
simulate_anova_data, 486
simulate_longitudinal_gompertz, 487
simulate_longitudinal_logistic, 491
simulate_longitudinal_negative_exponential, 495
simulate_longitudinal_polynomial, 498
simulate_longitudinal_richards, 504
simulate_regression_data, 508
* **datasets**
bessel_errors, 36
depression_bdi, 204
diagnosis_agreement, 213
drinks_trial, 220
holzinger_swineford, 269
prime_time_achievement, 407
pygmalion, 423
teacher_expectancy, 749
test_market, 751
* **descriptive statistics**
descriptives, 205
kurtosis, 290
skewness, 510
* **design utilities**
design_consequences, 207
design_effect, 211
effects_coding, 227
helmert_coding, 268
is_orthogonal_set, 286
orthogonal_polynomial, 366
* **design**
adjusted_means, 12
ancova, 18
anova_within, 23
anova_within_two_way, 24
bryant_paulson, 41
ci_c, 56
ci_c_ancova, 59

ci_c_ancova_bp, 62
ci_correlation, 64
ci_eta_squared, 73
ci_eta_squared_generalized, 77
ci_eta_squared_partial, 80
ci_games_howell, 83
ci_nc_chisq, 88
ci_nc_F, 90
ci_omega_squared, 95
ci_pvaf, 99
ci_rc, 104
ci_rmsea, 109
ci_sc, 111
ci_sc_ancova, 113
ci_sm, 118
ci_src, 127
ci_srsnr, 129
compare_cov_structures, 151
contrast_adjusted, 156
contrast_test, 158
convert_cor_cov, 161
convert_d_or, 163
convert_d_r, 164
convert_R2, 169
convert_r_Z, 168
convert_t_smd, 170
convert_Z_r, 173
correction_for_attenuation, 174
cv, 184
cv_bonferroni_f, 186
cv_bryant_paulson, 188
cv_dunnett, 192
cv_scheffe, 196
cv_smm, 198
cv_tukey_hsd, 201
design_consequences, 207
design_effect, 211
effects_coding, 227
eta_squared, 238
eta_squared_generalized, 241
eta_squared_partial, 245
factorial_anova, 256
helmert_coding, 268
is_orthogonal_set, 286
manova_split_plot, 296
mixed_anova, 331
mr_cv, 349
mr_smd, 351
omega_squared, 360
omega_squared_partial, 363
orthogonal_polynomial, 366
pairwise_within, 369
power_density_equivalence_md, 397
power_equivalence_c, 398
power_equivalence_md, 401
power_equivalence_md_plot, 403
responder_analysis, 469
sd_unbiased, 471
simple_effects_AB, 475
simulate_ancova_data, 480
simulate_ancova_factorial_data, 483
simulate_anova_data, 486
simulate_longitudinal_polynomial, 498
simulate_regression_data, 508
ss_aipe_c, 518
ss_aipe_c_ancova, 520
ss_aipe_c_ancova_sensitivity, 522
ss_aipe_c_sensitivity, 524
ss_aipe_cliff_delta, 526
ss_aipe_cliff_delta_sensitivity, 528
ss_aipe_composite_sem, 530
ss_aipe_cv, 547
ss_aipe_cv_sensitivity, 548
ss_aipe_equivalence_r, 551
ss_aipe_equivalence_r_sensitivity, 553
ss_aipe_equivalence_smd, 556
ss_aipe_equivalence_smd_sensitivity, 558
ss_aipe_icc, 560
ss_aipe_icc_sensitivity, 563
ss_aipe_indirect_effect, 566
ss_aipe_indirect_effect_sensitivity, 571
ss_aipe_mixed_effects, 573
ss_aipe_mixed_effects_sensitivity, 575
ss_aipe_omega_squared, 577
ss_aipe_omega_squared_sensitivity, 580
ss_aipe_partial_r, 582
ss_aipe_partial_r_sensitivity, 585
ss_aipe_pcm, 587

ss_aipe_pcm_sensitivity, 590
ss_aipe_r, 592
ss_aipe_R2, 596
ss_aipe_R2_sensitivity, 599
ss_aipe_r_sensitivity, 594
ss_aipe_rc, 602
ss_aipe_rc_sensitivity, 604
ss_aipe_reg_coef, 607
ss_aipe_reg_coef_sensitivity, 611
ss_aipe_reliability, 614
ss_aipe_reliability_sensitivity, 618
ss_aipe_rmsea, 620
ss_aipe_sc, 624
ss_aipe_sc_ancova, 626
ss_aipe_sc_ancova_sensitivity, 628
ss_aipe_sc_sensitivity, 631
ss_aipe_sem_path, 633
ss_aipe_sem_path_sensitivity, 636
ss_aipe_semipartial_r, 639
ss_aipe_semipartial_r_sensitivity, 641
ss_aipe_sm, 643
ss_aipe_sm_sensitivity, 645
ss_aipe_smd, 647
ss_aipe_smd_sensitivity, 649
ss_aipe_src, 652
ss_aipe_src_sensitivity, 655
ss_power_c, 657
ss_power_c_ancova, 659
ss_power_composite_ancova, 661
ss_power_composite_ancova_2group, 665
ss_power_composite_anova, 672
ss_power_composite_factorial_ancova, 675
ss_power_composite_factorial_ancova_het, 679
ss_power_composite_factorial_anova, 683
ss_power_composite_sem, 686
ss_power_contrast, 691
ss_power_equivalence_c, 694
ss_power_factorial_ancova, 696
ss_power_factorial_anova, 699
ss_power_indirect_effect, 701
ss_power_mixed_effects, 703
ss_power_one_way_anova, 705
ss_power_pcm, 707
ss_power_r, 711
ss_power_R2, 713
ss_power_R2_sensitivity, 716
ss_power_rc, 719
ss_power_reg_coef, 723
ss_power_reg_coef_sensitivity, 727
ss_power_rm_anova, 730
ss_power_sc, 732
ss_power_sem, 734
ss_power_smd, 736
ss_power_split_plot_anova, 739
ss_seq_c, 741
ss_seq_c_sensitivity, 744
var_icc, 773
var_partial_r, 780
var_R2, 784
var_semipartial_r, 785

∗ **distribution**

bryant_paulson, 41
convert_F_chisq, 165
cv_bonferroni_f, 186
cv_chisq, 191
cv_f, 194
moments_nc_chisq, 344
moments_nc_F, 345
moments_nc_t, 347

∗ **effect size estimates**

cles, 133
cliff_delta, 135
correction_for_attenuation, 174
eta_squared, 238
eta_squared_generalized, 241
eta_squared_partial, 245
expected_partial_r, 247
expected_r, 249
expected_smd, 254
nnt_from_smd, 354
omega_squared, 360
omega_squared_partial, 363
probability_of_superiority_paired, 418
proportion_of_superiority, 422
responder_analysis, 469
smd_trimmed, 516

∗ **equivalence testing**

equivalence_c, 230
equivalence_r, 234

equivalence_smd, 236
plot_equivalence, 375
power_density_equivalence_md, 397
power_equivalence_c, 398
power_equivalence_md, 401
power_equivalence_md_plot, 403
ss_power_equivalence_c, 694
∗ **hplot**
plot_cfa_k, 371
plot_ci, 372
plot_equivalence, 375
plot_forest, 377
plot_irt_information, 379
plot_mediation_mbco, 381
plot_R2, 384
plot_randomization_test, 386
plot_regions_of_significance, 387
plot_smd, 389
plot_trajectories, 392
plot_trajectories_fitted, 394
power_equivalence_md_plot, 403
∗ **htest**
ancova, 18
anova_within, 23
anova_within_two_way, 24
bayes_independent_t, 29
bayes_one_sample_t, 32
bayes_paired_t, 34
ci_c_ancova_bp, 62
ci_cv, 68
ci_dunnett, 70
ci_eigenvalue, 72
ci_eta_squared, 73
ci_eta_squared_generalized, 77
ci_eta_squared_partial, 80
ci_games_howell, 83
ci_mahalanobis, 85
ci_nc_t, 92
ci_omega_squared, 95
ci_proportion, 97
ci_R2, 101
ci_reg_coef, 106
ci_scheffe, 116
ci_smd, 120
ci_smd_c, 123
ci_tukey_kramer, 131
cles, 133
cliff_delta, 135
cohen_kappa, 140
combine_p, 145
compare_cov_structures, 151
contrast_adjusted, 156
contrast_test, 158
correlations_test, 177
cv, 184
cv_bonferroni_f, 186
cv_bryant_paulson, 188
cv_chisq, 191
cv_dunnett, 192
cv_f, 194
cv_scheffe, 196
cv_smm, 198
cv_tukey_hsd, 201
dunn_test, 223
epsilon_corrections, 229
equivalence_c, 230
equivalence_r, 234
equivalence_smd, 236
eta_squared, 238
eta_squared_generalized, 241
eta_squared_partial, 245
expected_partial_r, 247
expected_r, 249
expected_smd, 254
factorial_anova, 256
fleiss_kappa, 259
gwet_ac, 266
icc, 274
icc_lmer, 276
krippendorff_alpha, 287
limits_of_agreement, 291
lin_ccc, 294
manova_split_plot, 296
mauchly_test, 298
meta_contrast, 323
mixed_anova, 331
mr_cv, 349
mr_smd, 351
nnt_from_smd, 354
obrien_test, 358
omega_squared, 360
omega_squared_partial, 363
pairwise_within, 369
power_fisher_exact, 405
probability_of_superiority_paired, 418

procrustes_phi, 420
proportion_of_superiority, 422
R2_mixed_effects, 426
randomization_test, 431
randomization_test_paired, 435
regions_of_significance, 438
reliability, 441
reliability_alpha, 445
reliability_H, 452
reliability_kr20, 454
reliability_omega, 457
reliability_omega_categorical, 465
responder_analysis, 469
sd_unbiased, 471
signal_to_noise_R2, 473
simple_effects_AB, 475
smd, 512
smd_c, 514
smd_trimmed, 516
ss_aipe_c_sensitivity, 524
ss_aipe_cliff_delta_sensitivity, 528
ss_aipe_cv, 547
ss_aipe_cv_sensitivity, 548
ss_aipe_equivalence_r_sensitivity, 553
ss_aipe_equivalence_smd_sensitivity, 558
ss_aipe_icc_sensitivity, 563
ss_aipe_indirect_effect_sensitivity, 571
ss_aipe_omega_squared_sensitivity, 580
ss_aipe_partial_r_sensitivity, 585
ss_aipe_r_sensitivity, 594
ss_aipe_reliability_sensitivity, 618
ss_aipe_semipartial_r_sensitivity, 641
ss_aipe_sm_sensitivity, 645
ss_aipe_smd_sensitivity, 649
ss_power_c, 657
ss_power_c_ancova, 659
ss_power_composite_ancova, 661
ss_power_composite_ancova_2group, 665
ss_power_composite_anova, 672
ss_power_composite_factorial_ancova, 675
ss_power_composite_factorial_ancova_het, 679
ss_power_composite_factorial_anova, 683
ss_power_composite_sem, 686
ss_power_factorial_anova, 699
ss_power_mixed_effects, 703
ss_power_one_way_anova, 705
ss_power_r, 711
ss_power_rm_anova, 730
ss_power_sc, 732
ss_power_smd, 736
ss_power_split_plot_anova, 739
summary_t_test, 746
var_alpha, 767
var_cv, 769
var_ete, 770
var_indirect_effect, 775
var_omega_squared, 777
var_r, 782
var_smd, 787
var_smd_trimmed, 789
vargha_delaney_A, 791
variance_components_mls, 794
welch_t, 796

∗ **hypothesis tests**

adjusted_means, 12
ancova, 18
anova_within, 23
ci_dunnett, 70
ci_scheffe, 116
ci_tukey_kramer, 131
compare_cov_structures, 151
contrast_test, 158
correlations_test, 177
equivalence_r, 234
equivalence_smd, 236
factorial_anova, 256
manova_split_plot, 296
mauchly_test, 298
mixed_anova, 331
obrien_test, 358
pairwise_within, 369
randomization_test, 431
randomization_test_paired, 435
regions_of_significance, 438
simple_effects_AB, 475

summary_t_test, 746
welch_t, 796
* **mediation**
mediate, 312
mediation_mbco, 315
ss_power_indirect_effect, 701
* **meta-analysis**
combine_p, 145
meta_contrast, 323
meta_es, 325
meta_r, 327
meta_smd, 329
plot_forest, 377
* **misc**
mr_cv, 349
mr_smd, 351
* **mixed models**
icc_lmer, 276
manova_split_plot, 296
mixed_anova, 331
R2_mixed_effects, 426
R2_mixed_effects_decomposition, 429
ss_aipe_mixed_effects, 573
ss_aipe_mixed_effects_sensitivity, 575
ss_power_mixed_effects, 703
ss_power_split_plot_anova, 739
* **mixed-effects R-squared**
R2_mixed_effects_decomposition, 429
* **models**
adjusted_means, 12
analysis_of_change, 15
ci_cv, 68
ci_nc_t, 92
ci_smd_c, 123
irt_grm, 278
measurement_alignment, 300
measurement_invariance, 306
mediate, 312
mediation_mbco, 315
meta_es, 325
meta_r, 327
meta_smd, 329
mlmr, 333
mlmr_mv, 339
R2_mixed_effects_decomposition, 429
regions_of_significance, 438
signal_to_noise_R2, 473
smd, 512
smd_c, 514
* **multilevel R-squared**
R2_mixed_effects_decomposition, 429
* **multivariate and latent variable methods**
average_variance_extracted, 27
bifactor_indices, 39
cfa_1, 44
cfa_2, 47
cfa_k, 50
ci_eigenvalue, 72
common_method_marker, 147
common_method_single_factor, 149
dmacs, 215
ecvi, 226
htmt, 272
irt_grm, 278
irt_information, 282
measurement_alignment, 300
measurement_invariance, 306
procrustes_phi, 420
simple_structure, 478
* **multivariate**
average_variance_extracted, 27
bifactor_indices, 39
cfa_1, 44
cfa_2, 47
cfa_k, 50
ci_mahalanobis, 85
ci_nc_chisq, 88
ci_nc_F, 90
ci_R2, 101
common_method_marker, 147
common_method_single_factor, 149
content_validity_index, 153
convert_R2, 169
correlations_test, 177
cov_sem, 180
covmat_from_cfa, 182
descriptives, 205
design_effect, 211
dmacs, 215
ecvi, 226
htmt, 272

icc, 274
irt_grm, 278
irt_information, 282
measurement_alignment, 300
measurement_invariance, 306
mlmr_mv, 339
reliability, 441
reliability_alpha, 445
reliability_kr20, 454
reliability_omega, 457
reliability_omega_categorical, 465
simple_structure, 478
ss_aipe_composite_sem, 530
ss_aipe_mixed_effects_sensitivity, 575
ss_aipe_pcm, 587
ss_aipe_pcm_sensitivity, 590
ss_aipe_sem_path, 633
ss_aipe_sem_path_sensitivity, 636
ss_power_composite_sem, 686
ss_power_sem, 734

∗ **noncentral distribution confidence intervals**
ci_nc_chisq, 88
ci_nc_F, 90
ci_nc_t, 92

∗ **noncentral distribution moments**
moments_nc_chisq, 344
moments_nc_F, 345
moments_nc_t, 347

∗ **nonparametric**
dunn_test, 223
randomization_test, 431
vargha_delaney_A, 791

∗ **package**
DMAR-package, 10

∗ **parameterization conversions**
convert_cor_cov, 161
convert_d_or, 163
convert_d_r, 164
convert_F_chisq, 165
convert_R2, 169
convert_r_Z, 168
convert_t_smd, 170
convert_z_normal, 172
convert_Z_r, 173

∗ **plotting**
plot_cfa_k, 371
plot_ci, 372
plot_equivalence, 375
plot_forest, 377
plot_irt_information, 379
plot_mediation_mbco, 381
plot_R2, 384
plot_randomization_test, 386
plot_regions_of_significance, 387
plot_smd, 389
plot_trajectories, 392
plot_trajectories_fitted, 394
power_equivalence_md_plot, 403

∗ **probability of superiority**
cles, 133
vargha_delaney_A, 791

∗ **regression**
analysis_of_change, 15
ci_correlation, 64
ci_nc_chisq, 88
ci_nc_F, 90
ci_R2, 101
mlmr, 333
mlmr_mv, 339
var_partial_r, 780
var_semipartial_r, 785

∗ **reliability**
cohen_kappa, 140
diagnosis_agreement, 213
fleiss_kappa, 259
icc, 274
reliability, 441
reliability_alpha, 445
reliability_H, 452
reliability_kr20, 454
reliability_omega, 457
reliability_omega_categorical, 465

∗ **sample size for power**
power_fisher_exact, 405
ss_aipe_mixed_effects, 573
ss_power_c, 657
ss_power_c_ancova, 659
ss_power_composite_ancova, 661
ss_power_composite_ancova_2group, 665
ss_power_composite_anova, 672
ss_power_composite_factorial_ancova, 675
ss_power_composite_factorial_ancova_het,

679
ss_power_composite_factorial_anova, 683
ss_power_composite_sem, 686
ss_power_contrast, 691
ss_power_equivalence_c, 694
ss_power_factorial_ancova, 696
ss_power_factorial_anova, 699
ss_power_indirect_effect, 701
ss_power_mixed_effects, 703
ss_power_one_way_anova, 705
ss_power_pcm, 707
ss_power_r, 711
ss_power_R2, 713
ss_power_R2_sensitivity, 716
ss_power_rc, 719
ss_power_reg_coef, 723
ss_power_reg_coef_sensitivity, 727
ss_power_rm_anova, 730
ss_power_sc, 732
ss_power_sem, 734
ss_power_smd, 736
ss_power_split_plot_anova, 739
∗ **sequential estimation**
ss_seq_c, 741
ss_seq_c_sensitivity, 744
∗ **univar**
ci_smd, 120
descriptives, 205
kurtosis, 290
skewness, 510
var_icc, 773
∗ **variance utilities**
var_alpha, 767
var_cv, 769
var_ete, 770
var_indirect_effect, 775
var_omega_squared, 777
var_r, 782
var_smd, 787
var_smd_trimmed, 789
∗ **within-subjects analysis**
anova_within, 23
anova_within_two_way, 24
epsilon_corrections, 229
mauchly_test, 298
pairwise_within, 369
plot_trajectories_fitted, 394

adjusted_means, 12
adjusted_means(), *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *235*, *237*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *441*, *477*, *748*, *797*
alpha, *449*, *451*, *455, 456*
analysis_of_change, 15
ancova, *14*, 18, *64*, *258*, *426*, *441*, *670*, *698*, *752*, *772*
ancova(), *14*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *235*, *237*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *441*, *477*, *748*, *797*
Anova, *296, 297*
anova, *95*, *152*, *361*, *364*, *778*
anova.mlmr, 20, *22*
anova.mlmr_mv, 22
anova_within, 23, *26*, *230*, *299, 300*, *332*, *370*
anova_within(), *14*, *20*, *26*, *71*, *117*, *132*, *152*, *160*, *179*, *230*, *235*, *237*, *258*, *297*, *300*, *332*, *359*, *370*, *396*, *435*, *437*, *441*, *477*, *748*, *797*
anova_within_two_way, 24, *297*, *332*
anova_within_two_way(), *24*, *230*, *300*, *370*, *396*
aov, *12, 13*, *24*, *70*, *73, 74*, *77*, *81*, *83*, *95*, *116*, *132*, *156–159*, *224*, *238*, *241, 242*, *245*, *360, 361*, *363*, *367*, *475*, *778*
as_kable *(*dmar_output_helpers*)*, 218
average_variance_extracted, 27, *49*, *55*, *274*, *480*
average_variance_extracted(), *40*, *46*, *49*, *55*, *73*, *148*, *150*, *218*, *227*, *274*, *281*, *285*, *304*, *310*, *421*, *480*

bartlett.test, *359*
bayes_independent_t, 29, *34*, *36*
bayes_independent_t(), *34*, *36*
bayes_one_sample_t, *29–31*, 32, *34–36*
bayes_one_sample_t(), *31*, *36*
bayes_paired_t, *31*, 34, *34*
bayes_paired_t(), *31*, *34*
bessel_errors, 36
bifactor_indices, 39
bifactor_indices(), *28*, *46*, *49*, *55*, *73*, *148*, *150*, *218*, *227*, *274*, *281*, *285*, *304*, *310*, *421*, *480*
binom.test, *154*
bootMer, *427, 428*
bryant_paulson, 41, *62*, *64*, *752*

cfa, *45*, *48*, *52*, *281*, *307*
cfa_1, *27, 28*, 44, *49*, *52*, *55*, *175*, *183, 184*, *218*, *281*, *304*, *306*, *310*, *444*, *451*, *459*, *464*, *616, 617*
cfa_1(), *28*, *40*, *49*, *55*, *73*, *148*, *150*, *218*, *227*, *274*, *281*, *285*, *304*, *310*, *421*, *480*
cfa_2, *46*, 47, *52*, *55*
cfa_2(), *28*, *40*, *46*, *55*, *73*, *148*, *150*, *218*, *227*, *274*, *281*, *285*, *304*, *310*, *421*, *480*
cfa_k, *44*, *46, 47*, *49*, 50, *263, 264*, *310*, *371, 372*, *753*
cfa_k(), *28*, *40*, *46*, *49*, *73*, *148*, *150*, *218*, *227*, *274*, *281*, *285*, *304*, *310*, *421*, *480*
ci_c, 56, *61*, *113*, *231*, *233*, *367, 368*, *519*, *524*, *526*, *659*
ci_c(), *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_c_ancova, *20*, 59, *62–64*, *113*, *115*, *158*, *205*, *482*, *521*, *661*, *670*
ci_c_ancova(), *58*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_c_ancova_bp, *43*, *59–61*, 62, *157*, *189, 190*, *752*
ci_c_ancova_bp(), *58*, *61*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_correlation, *58*, *61*, 64, *64*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_cv, 68, *92*, *184, 185*, *351*, *550*, *769, 770*
ci_cv(), *58*, *61*, *64*, *67*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_dunnett, *14*, 70, *84*, *117*, *132*, *758*
ci_dunnett(), *14*, *20*, *24*, *117*, *132*, *152*, *160*, *179*, *235*, *237*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *441*, *477*, *748*, *797*
ci_eigenvalue, 72
ci_eigenvalue(), *28*, *40*, *46*, *49*, *55*, *148*, *150*, *218*, *227*, *274*, *281*, *285*, *304*, *310*, *421*, *480*
ci_eta_squared, 73, *79*, *81, 82*, *239, 240*, *361, 362*, *758*
ci_eta_squared(), *58*, *61*, *64*, *67*, *69*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_eta_squared_generalized, 77, *243*, *758*
ci_eta_squared_generalized(), *58*, *61*, *64*, *67*, *69*, *76*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_eta_squared_partial, *19*, *76*, 80, *246*, *257, 258*, *365*, *477*, *758*
ci_eta_squared_partial(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_games_howell, 83, *224, 225*, *758*
ci_mahalanobis, 85
ci_mahalanobis(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_nc_chisq, 88, *90*, *92*, *94*, *110*, *167*, *192*, *226, 227*, *345*
ci_nc_chisq(), *92*, *94*
ci_nc_F, *66*, *74*, *76*, *78*, *86–88*, 90, *90*, *92*, *94*, *96, 97*, *99, 100*, *102, 103*, *126*, *131*, *167*, *170*, *195, 196*, *347*, *476, 477*, *700*, *707*, *716*, *721*, *725*
ci_nc_F(), *90*, *94*
ci_nc_t, *67*, *88*, *90*, 92, *92*, *106*, *109*, *113*, *119*, *121, 122*, *124*, *129*, *170*, *514, 515*, *517, 518*, *598*, *602*, *604*, *610*, *625*, *633*, *644*, *649*, *654*, *738*
ci_nc_t(), *90*, *92*
ci_omega_squared, *19*, *74*, *76*, *78*, 95, *239, 240*, *243*, *257, 258*, *361, 362*, *364, 365*, *373, 374*, *577–581*, *758*, *778, 779*
ci_omega_squared(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_proportion, 97, *140*, *470, 471*

ci_pvaf, *74*, *76*, *78*, *90*, *96, 97*, 99, *240*, *362*, *758*
ci_pvaf(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_R, *374*, *385*
ci_R *(*ci_correlation*)*, 64
ci_r, *165*, *169*, *174, 175*, *179*, *235*, *252*, *276*, *551*, *553, 554*, *583*, *713*, *758*, *781–783*
ci_r *(*ci_correlation*)*, 64
ci_R2, *66, 67*, *87*, *90*, *92*, 101, *170*, *254*, *373, 374*, *385*, *474*, *510*, *598*, *602*, *716*, *719*, *766*, *785*
ci_R2(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_rc, 104, *106*, *109*, *127*, *129*, *338*
ci_rc(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_reg_coef, *104, 105*, 106, *106*, *127–129*, *338*, *607*, *609*, *613*, *657*, *758*, *787*
ci_reg_coef(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_rmsea, 109, *226, 227*, *621–623*
ci_rmsea(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_sc, *58*, 111, *115*, *624, 625*, *733*
ci_sc(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_sc_ancova, *20*, *61*, 113, *205*, *521*, *670*, *698*
ci_sc_ancova(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *119*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_scheffe, *71*, *84*, 116, *132*, *287*, *758*
ci_scheffe(), *14*, *20*, *24*, *71*, *132*, *152*, *160*, *179*, *235*, *237*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *441*, *477*, *748*, *797*
ci_sm, *33, 34*, *113*, 118, *644*
ci_sm(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *122*, *124*, *126*, *129*, *131*, *158*, *391*
ci_smd, *31*, *33*, *87*, *92*, *94*, *113*, 120, *124*, *135*, *171*, *222*, *237*, *239*, *256*, *330*, *353*, *356*, *361*, *373, 374*, *391*, *422, 423*, *433–435*, *514*, *517, 518*, *557, 558*, *649*, *738*, *788, 789*, *793*, *797*
ci_smd(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *124*, *126*, *129*, *131*, *158*, *391*
ci_smd_c, *92*, *94*, *113*, *121, 122*, 123, *514*, *649*, *758*
ci_smd_c(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *126*, *129*, *131*, *158*, *391*
ci_snr, *90*, 125, *131*, *138*
ci_snr(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *129*, *131*, *158*, *391*
ci_src, *104–106*, *109*, *113*, 127, *338*
ci_src(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *131*, *158*, *391*
ci_srsnr, *126*, 129, *138*
ci_srsnr(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *158*, *391*
ci_tukey_kramer, *71*, *83, 84*, *117*, 131, *224, 225*, *758*
ci_tukey_kramer(), *14*, *20*, *24*, *71*, *117*, *152*, *160*, *179*, *235*, *237*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *441*, *477*, *748*, *797*
cles, 133, *136, 137*, *355, 356*, *419*, *422, 423*, *433–435*
cles(), *137*, *176*, *240*, *243*, *246*, *248*, *252*, *256*, *356*, *362*, *365*, *419*, *423*, *471*, *518*
cliff_delta, 135, *222*, *225*, *418, 419*, *433–435*, *471*, *528–530*
cliff_delta(), *135*, *176*, *240*, *243*, *246*, *248*, *252*, *256*, *356*, *362*, *365*, *419*, *423*,

471, *518*
cohen_f, 137, *140*
cohen_h, 139
cohen_kappa, 140, *214*, *260, 261*, *267*, *287*, *289*, *793*
cohen_kappa(), *214*, *261*, *276*, *444*, *451*, *454*, *456*, *464*, *468*
combine_p, 145, *324*, *327*, *750*
combine_p(), *324*, *327, 328*, *330*, *378*
common_method_marker, 147, *149, 150*
common_method_marker(), *28*, *40*, *46*, *49*, *55*, *73*, *150*, *218*, *227*, *274*, *281*, *285*, *304*, *310*, *421*, *480*
common_method_single_factor, *148*, 149
common_method_single_factor(), *28*, *40*, *46*, *49*, *55*, *73*, *148*, *218*, *227*, *274*, *281*, *285*, *304*, *310*, *421*, *480*
compare_cov_structures, 151, *310*
compare_cov_structures(), *14*, *20*, *24*, *71*, *117*, *132*, *160*, *179*, *235*, *237*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *441*, *477*, *748*, *797*
content_validity_index, 153
content_validity_index(), *267*, *277*, *289*, *294, 295*, *428*, *795*
contr.helmert, *269*
contr.poly, *366–368*
contr.sum, *228*, *257*
contrast_adjusted, *13, 14*, 156, *187*, *224*
contrast_adjusted(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*, *391*
contrast_test, 158, *158*, *194*, *197*, *222*, *233*, *324*, *367, 368*, *477*, *487*, *693*, *759*
contrast_test(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *179*, *235*, *237*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *441*, *477*, *748*, *797*
convert_chisq_F *(*convert_F_chisq*)*, 165
convert_cor_cov, 161
convert_cor_cov(), *163*, *165*, *167*, *169–172*, *174*
convert_d_or, 163, *165*
convert_d_or(), *162*, *165*, *167*, *169–172*, *174*
convert_d_r, *163*, 164
convert_d_r(), *162, 163*, *167*, *169–172*, *174*
convert_delta_lambda *(*convert_t_smd*)*, 170
convert_F_chisq, 165
convert_F_chisq(), *162, 163*, *165*, *169–172*, *174*
convert_f_R2 *(*convert_R2*)*, 169
convert_lambda_delta *(*convert_t_smd*)*, 170
convert_lambda_R2 *(*convert_R2*)*, 169
convert_or_d, *165*
convert_or_d *(*convert_d_or*)*, 163
convert_R2, *162, 163*, *165*, *167*, 169, *169*, *171, 172*, *174*
convert_R2_f, *473, 474*
convert_R2_f *(*convert_R2*)*, 169
convert_R2_lambda *(*convert_R2*)*, 169
convert_r_d, *163*
convert_r_d *(*convert_d_r*)*, 164
convert_r_Z, *67*, 168, *173, 174*, *176*, *328*, *594*, *713*, *781*
convert_r_Z(), *162, 163*, *165*, *167*, *170–172*, *174*
convert_t_smd, *162, 163*, *165*, *167*, *169*, 170, *170*, *172*, *174*
convert_z_normal, 172
convert_z_normal(), *162, 163*, *165*, *167*, *169–171*, *174*
convert_Z_r, *67*, *168, 169*, 173, *176*, *328*, *713*
convert_Z_r(), *162, 163*, *165*, *167*, *169–172*
cor, *179*, *207*, *252*, *421*
cor.test, *178, 179*, *252*
correction_for_attenuation, 174, *273, 274*, *327, 328*
correction_for_attenuation(), *135*, *137*, *240*, *243*, *246*, *248*, *252*, *256*, *356*, *362*, *365*, *419*, *423*, *471*, *518*
correlations_test, 177, *262*, *592*, *594*
correlations_test(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *235*, *237*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *441*, *477*, *748*, *797*
cov2cor, *150*, *206*
cov_sem, 180, *531*, *534*, *634, 635*, *637, 638*, *687*, *690*
covmat_from_cfa, 182, *182*
cv, *69*, 184, *351*, *472*, *548*, *550*
cv_bonferroni_f, 186, *196*
cv_bonferroni_f(), *190*, *192*, *194*, *196, 197*,

199, *201, 202*, *204*
cv_bryant_paulson, 188
cv_bryant_paulson(), *187*, *192*, *194*, *196, 197*, *199*, *201, 202*, *204*
cv_chisq, *167*, 191, *196*
cv_chisq(), *187*, *190*, *194*, *196, 197*, *199*, *201, 202*, *204*
cv_dunnett, *71*, *187*, *189*, 192, *197*, *199*, *202*
cv_dunnett(), *187*, *190*, *192*, *196, 197*, *199*, *201, 202*, *204*
cv_f, *167*, *186, 187*, *192*, 194
cv_f(), *187*, *190*, *192*, *194*, *197*, *199*, *201, 202*, *204*
cv_scheffe, *117*, *157*, *187*, *190*, *194*, 196, *196*, *199*
cv_scheffe(), *187*, *190*, *192*, *194*, *196*, *199*, *201, 202*, *204*
cv_smm, *84*, *189*, *194*, *197*, 198, *202*
cv_smm(), *187*, *190*, *192*, *194*, *196, 197*, *201, 202*, *204*
cv_t, *187*, *189*, *191, 192*, *194–197*, *199*, 200, *202*, *693*
cv_t(), *187*, *190*, *192*, *194*, *196, 197*, *199*, *202*, *204*
cv_tukey_hsd, *83, 84*, *132*, *160*, *187–190*, *194*, *197*, *199*, 201
cv_tukey_hsd(), *187*, *190*, *192*, *194*, *196, 197*, *199*, *201*, *204*
cv_z, *172*, *202*, 203
cv_z(), *187*, *190*, *192*, *194*, *196, 197*, *199*, *201, 202*

dbryant_paulson *(*bryant_paulson*)*, 41
dchisq, *345*
depression_bdi, 204
descriptives, *179*, 205, *276*, *290, 291*, *511*
descriptives(), *291*, *511*
design_consequences, 207, *519*, *521*, *524*, *526*, *528*, *530*, *539*, *545*, *548*, *550*, *557*, *560*, *562*, *565*, *570*, *572*, *574*, *577*, *579*, *581*, *584*, *586*, *589*, *591*, *594*, *596*, *598*, *602*, *604*, *607*, *610*, *613*, *617*, *619*, *621*, *625*, *628*, *631*, *633*, *635*, *638*, *641*, *643, 644*, *647*, *649*, *651*, *654*, *657*, *659*, *661*, *693*, *698*, *700*, *702*, *705*, *707*, *710*, *713*, *716*, *719*, *721*, *725*, *729*, *732, 733*, *736*, *738*, *741*
design_consequences(), *212*, *228*, *269*, *287*, *368*
design_effect, 211
design_effect(), *210*, *228*, *269*, *287*, *368*
df, *347*
diagnosis_agreement, *143*, 213, *261*, *276*, *444*, *451*, *454*, *456*, *464*, *468*
difftime, *357*
dmacs, 215
dmacs(), *28*, *40*, *46*, *49*, *55*, *73*, *148*, *150*, *227*, *274*, *281*, *285*, *304*, *310*, *421*, *480*
DMAR *(*DMAR-package*)*, 10
dmar *(*DMAR-package*)*, 10
DMAR-package, 10
dmar_output_helpers, 218
dmar_tbl, *14*, *20*, *84*, *160*, *177*, *218*, *220*, *225*, *258*, *262, 263*, *756*, *760*
dmar_tidiers, *84*, *160*
dmar_tidiers *(*tidy.dmar_contrast_test*)*, 756
drinks_trial, 220
dt, *348*
dunn_test, *187*, 223

ecvi, 226
ecvi(), *28*, *40*, *46*, *49*, *55*, *73*, *148*, *150*, *218*, *274*, *281*, *285*, *304*, *310*, *421*, *480*
effects_coding, 227, *269*, *287*, *368*
effects_coding(), *210*, *212*, *269*, *287*, *368*
eigen, *73*
epsilon_corrections, *24*, *26*, 229, *299, 300*
epsilon_corrections(), *24*, *26*, *300*, *370*, *396*
equivalence_c, 230, *375, 376*, *400*, *696*, *744*
equivalence_c(), *235*, *238*, *376*, *398*, *400*, *402*, *404*, *696*
equivalence_r, *233*, 234, *237*, *551*, *553–555*
equivalence_r(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *233*, *238*, *258*, *297*, *300*, *332*, *359*, *370*, *376*, *398*, *400*, *402*, *404*, *435*, *437*, *441*, *477*, *696*, *748*, *797*
equivalence_smd, *233*, *235*, 236, *557*, *559, 560*
equivalence_smd(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *233*, *235*, *258*, *297*, *300*, *332*, *359*, *370*, *376*, *398*, *400*, *402*, *404*, *435*, *437*, *441*, *477*, *696*, *748*, *797*

eta_squared, *76*, 238, *243*, *245*, *246*, *361–363*
eta_squared(), *135*, *137*, *176*, *243*, *246*, *248*, *252*, *256*, *356*, *362*, *365*, *419*, *423*, *471*, *518*
eta_squared_generalized, *77*, *79*, *239*, 241
eta_squared_generalized(), *135*, *137*, *176*, *240*, *246*, *248*, *252*, *256*, *356*, *362*, *365*, *419*, *423*, *471*, *518*
eta_squared_partial, *82*, *240*, *243*, 245, *363*, *365*, *477*
eta_squared_partial(), *135*, *137*, *176*, *240*, *243*, *248*, *252*, *256*, *356*, *362*, *365*, *419*, *423*, *471*, *518*
expand.grid, *13*, *157*, *483*, *484*
expected_partial_r, 247, *584*
expected_partial_r(), *135*, *137*, *176*, *240*, *243*, *246*, *252*, *256*, *356*, *362*, *365*, *419*, *423*, *471*, *518*
expected_r, *248*, 249, *256*, *783*
expected_r(), *135*, *137*, *176*, *240*, *243*, *246*, *248*, *256*, *356*, *362*, *365*, *419*, *423*, *471*, *518*
expected_R2, 253, *256*, *765*, *766*, *785*
expected_smd, *210*, 254, *329*, *348*, *788*, *789*
expected_smd(), *135*, *137*, *176*, *240*, *243*, *246*, *248*, *252*, *356*, *362*, *365*, *420*, *423*, *471*, *518*

facet_wrap, *393*
factanal, *149*
factor, *439*
factorial_anova, 256
factorial_anova(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *235*, *238*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *441*, *477*, *748*, *797*
fisher.test, *407*
fleiss_kappa, *143*, *155*, 259, *267*, *287*, *289*
fleiss_kappa(), *143*, *214*, *276*, *444*, *451*, *454*, *456*, *464*, *468*
format.dmar_tbl, *219*
format_p, *220*, 262, *416*, *418*
formula, *257*, *333*, *340*

ggsave, *393*
glance, *84*, *160*, *658*, *660*, *663*, *668*, *673*, *677*, *681*, *685*, *689*, *692*, *695*, *698*, *700*, *702*, *704*, *706*, *718*, *721*, *729*, *731*, *733*, *740*, *756*
glance.dmar_cfa_k, 263, *753*
glance.dmar_ci_anova *(*tidy.dmar_contrast_test*)*, 756
glance.dmar_ci_long *(*tidy.dmar_contrast_test*)*, 756
glance.dmar_ci_R2 *(*tidy.dmar_ci_R2*)*, 754
glance.dmar_ci_smd *(*tidy.dmar_ci_smd*)*, 755
glance.dmar_content_validity *(*tidy.dmar_contrast_test*)*, 756
glance.dmar_contrast_test *(*tidy.dmar_contrast_test*)*, 756
glance.dmar_dmacs *(*tidy.dmar_contrast_test*)*, 756
glance.dmar_measurement_alignment *(*tidy.dmar_contrast_test*)*, 756
glance.dmar_measurement_invariance *(*tidy.dmar_contrast_test*)*, 756
glance.dmar_mediation_mbco, 264
glance.dmar_post_hoc_ci *(*tidy.dmar_contrast_test*)*, 756
glance.dmar_R2_mixed_effects *(*tidy.dmar_R2_mixed_effects*)*, 761
glance.dmar_reliability, 265
glance.dmar_ss_aipe *(*tidy.dmar_contrast_test*)*, 756
glance.dmar_ss_power *(*tidy.dmar_contrast_test*)*, 756
glance.dmar_ss_power_sensitivity *(*tidy.dmar_contrast_test*)*, 756
glance.dmar_summary_t_test *(*summary_t_test_broom*)*, 748
glance.dmar_tbl *(*tidy.dmar_contrast_test*)*, 756
glance.dmar_welch_t *(*welch_t_broom*)*, 798
glance.mlmr, 265
glance.mlmr_mv *(*tidy.mlmr_mv*)*, 764
glht, *194*
gls, *151*, *152*
gwet_ac, *155*, 266
gwet_ac(), *155*, *277*, *289*, *294*, *295*, *428*, *795*

helmert_coding, *228*, 268, *287*, *368*
helmert_coding(), *210*, *212*, *228*, *287*, *368*
holzinger_swineford, 269
htmt, *27*, *28*, *45*, *49*, *52*, *55*, 272, *304*, *480*
htmt(), *28*, *40*, *46*, *49*, *55*, *73*, *148*, *150*, *218*, *227*, *281*, *285*, *304*, *310*, *421*, *480*

icc, *143*, *211*, *212*, *260*, *261*, 274, *277*, *289*, *562*, *563*, *565*, *773–775*, *795*
icc(), *143*, *214*, *261*, *444*, *451*, *454*, *456*, *464*, *468*
icc_lmer, *211*, *212*, 276, *426*, *428*, *430*, *577*
icc_lmer(), *155*, *267*, *289*, *294*, *295*, *298*, *332*, *428*, *430*, *575*, *577*, *705*, *741*, *795*
integrate, *42*, *193*, *199*
irt_grm, 278
irt_grm(), *28*, *40*, *46*, *49*, *55*, *73*, *148*, *150*, *218*, *227*, *274*, *285*, *304*, *310*, *421*, *480*
irt_information, 282, *379*, *380*
irt_information(), *28*, *40*, *46*, *49*, *55*, *73*, *148*, *150*, *218*, *227*, *274*, *281*, *304*, *310*, *421*, *480*
is_orthogonal_set, *228*, *269*, 286, *367*, *368*
is_orthogonal_set(), *210*, *212*, *228*, *269*, *368*

kable, *179*, *219*
knit_print.dmar_tbl *(*dmar_output_helpers*)*, 218
krippendorff_alpha, *267*, 287
krippendorff_alpha(), *155*, *267*, *277*, *294*, *295*, *428*, *795*
kruskal.test, *225*
kurtosis, 290, *511*
kurtosis(), *207*, *511*

lavaan, *46*, *49*, *52*, *55*, *335*, *338*, *341*
lavTestLRT, *20*, *22*, *55*, *308*
lavTestScore, *309*, *310*
lchoose, *154*
limits_of_agreement, 291, *295*
limits_of_agreement(), *155*, *267*, *277*, *289*, *295*, *428*, *795*
lin_ccc, 294, *294*
lin_ccc(), *155*, *267*, *277*, *289*, *294*, *428*, *795*
lm, *12*, *13*, *70*, *73*, *74*, *77*, *81*, *83*, *95*, *116*, *132*, *156–159*, *224*, *238*, *241*, *242*, *245*, *333*, *338*, *340*, *342*, *360*, *361*, *363*, *367*, *438*, *475*, *778*
lme, *427*
lmer, *25*, *276*, *277*, *331*, *332*, *427–429*
loa *(*limits_of_agreement*)*, 291
logLik, *152*

manova, *297*
manova_split_plot, *258*, 296
manova_split_plot(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *235*, *238*, *258*, *277*, *300*, *332*, *359*, *370*, *428*, *430*, *435*, *437*, *441*, *477*, *575*, *577*, *705*, *741*, *748*, *797*
mauchly.test, *299*
mauchly_test, *23*, *24*, *26*, *230*, 298
mauchly_test(), *14*, *20*, *24*, *26*, *71*, *117*, *132*, *152*, *160*, *179*, *230*, *235*, *238*, *258*, *297*, *332*, *359*, *370*, *396*, *435*, *437*, *441*, *477*, *748*, *797*
measurement_alignment, 300
measurement_alignment(), *28*, *40*, *46*, *49*, *55*, *73*, *148*, *150*, *218*, *227*, *274*, *281*, *285*, *310*, *421*, *480*
measurement_invariance, *55*, *217*, *218*, *300*, *304*, 305
measurement_invariance(), *28*, *40*, *46*, *49*, *55*, *73*, *148*, *150*, *218*, *227*, *274*, *281*, *285*, *304*, *421*, *480*
mediate, 312, *322*, *701*, *702*
mediate(), *322*, *702*
mediation_mbco, *264*, 315, *381–383*, *567*, *568*, *570*, *761*
mediation_mbco(), *314*, *702*
meta_contrast, *147*, 323, *327*, *750*
meta_contrast(), *147*, *327*, *328*, *330*, *378*
meta_es, *145*, *147*, *324*, 325, *328–330*, *377*, *378*
meta_es(), *147*, *324*, *328*, *330*, *378*
meta_r, *325*, 327, *327*, *378*
meta_r(), *147*, *324*, *327*, *330*, *378*
meta_smd, *145*, *147*, *325*, *327*, 329, *378*, *750*
meta_smd(), *147*, *324*, *327*, *328*, *378*
mixed_anova, *258*, 331
mixed_anova(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *235*, *238*, *258*, *277*, *297*, *298*, *300*, *359*, *370*, *428*, *430*, *435*, *437*, *441*, *477*, *575*, *577*, *705*, *741*, *748*, *797*
mlmr, *20*, *21*, *313*, 333, *340–342*, *764*
mlmr_mv, *22*, 339, *764*
model.matrix, *333*, *336*, *342*
model.syntax, *181*, *182*, *531*, *622*, *634–637*, *687*
moments_nc_chisq, 344

moments_nc_chisq(), *347, 348*
moments_nc_F, 345, *345*, *348*
moments_nc_F(), *345*, *348*
moments_nc_t, *345*, 347, *347*
moments_nc_t(), *345*, *347*
mr_cv, 349, *353*
mr_smd, 351, *351*, *742*, *744*
mvrnorm, *321*, *482*, *502*, *510*
mxSE, *318*

na.omit, *358*
nls, *16*
nnt_from_smd, *134, 135*, *137*, 354, *423*, *471*
nnt_from_smd(), *135*, *137*, *176*, *240*, *243*, *246*, *248*, *252*, *256*, *362*, *365*, *420*, *423*, *471*, *518*
nobs, *74*, *77*, *95*, *361*, *363*
now, 356

obrien_test, 358
obrien_test(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *235*, *238*, *258*, *297*, *300*, *332*, *370*, *435*, *437*, *441*, *477*, *748*, *797*
omega, *462*, *464*, *467, 468*
omega_squared, 360, *363–365*, *579*, *778, 779*
omega_squared(), *135*, *137*, *176*, *240*, *243*, *246*, *248*, *252*, *256*, *356*, *365*, *420*, *423*, *471*, *518*
omega_squared_partial, *20*, 363, *579*, *779*
omega_squared_partial(), *135*, *137*, *176*, *240*, *243*, *246*, *248*, *252*, *256*, *356*, *362*, *420*, *423*, *471*, *518*
oneway.test, *476*
optim, *302*
options, *250*
orthogonal_polynomial, 366
orthogonal_polynomial(), *210*, *212*, *228*, *269*, *287*

p.adjust, *159, 160*, *187*, *224, 225*, *476, 477*
pairwise.t.test, *160*, *370*
pairwise_within, 369
pairwise_within(), *14*, *20*, *24*, *26*, *71*, *117*, *132*, *152*, *160*, *179*, *230*, *235*, *238*, *258*, *297*, *300*, *332*, *359*, *396*, *435*, *437*, *441*, *477*, *748*, *797*
pbryant_paulson *(*bryant_paulson*)*, 41
plot, *667, 668*
plot.dmar_composite_power *(*ss_power_composite_ancova_2group*)*, 665
plot.dmar_composite_power_factorial *(*ss_power_composite_factorial_ancova*)*, 675
plot.dmar_composite_power_factorial_het *(*ss_power_composite_factorial_ancova_het*)*, 679
plot_cfa_k, *55*, 371
plot_cfa_k(), *374*, *376*, *378*, *380*, *383*, *385*, *387*, *389*, *391*, *394*, *396*, *404*
plot_ci, 372, *372*, *376*, *385*, *389*, *391*
plot_ci(), *372*, *376*, *378*, *380*, *383*, *385*, *387*, *389*, *391*, *394*, *396*, *404*
plot_equivalence, *233*, 375
plot_equivalence(), *233*, *235*, *238*, *372*, *374*, *378*, *380*, *383*, *385*, *387*, *389*, *391*, *394*, *396*, *398*, *400*, *402*, *404*, *696*
plot_forest, *327*, *330*, 377
plot_forest(), *147*, *324*, *327, 328*, *330*, *372*, *374*, *376*, *380*, *383*, *385*, *387*, *389*, *391*, *394*, *396*, *404*
plot_irt_information, *285*, 379
plot_irt_information(), *372*, *374*, *376*, *378*, *383*, *385*, *387*, *389*, *391*, *394*, *396*, *404*
plot_mediation_mbco, *320–322*, 381
plot_mediation_mbco(), *372*, *374*, *376*, *378*, *380*, *385*, *387*, *389*, *391*, *394*, *396*, *404*
plot_R2, *374*, 384, *391*
plot_R2(), *372*, *374*, *376*, *378*, *380*, *383*, *387*, *389*, *391*, *394*, *396*, *404*
plot_randomization_test, 386, *434, 435*
plot_randomization_test(), *372*, *374*, *376*, *378*, *380*, *383*, *385*, *389*, *391*, *394*, *396*, *404*
plot_regions_of_significance, 387, *440, 441*
plot_regions_of_significance(), *372*, *374*, *376*, *378*, *380*, *383*, *385*, *387*, *391*, *394*, *396*, *404*
plot_smd, *122*, *374*, *385*, 389, *514*
plot_smd(), *58*, *61*, *64*, *67*, *69*, *76*, *79*, *82*, *87*, *97*, *100*, *103*, *106*, *109, 110*, *113*, *115*, *119*, *122*, *124*, *126*, *129*, *131*,

158, *372*, *374*, *376*, *378*, *380*, *383*, *385*, *387*, *389*, *394*, *396*, *404*
plot_trajectories, *17*, 392, *395*, *396*, *489*, *490*, *493*, *496*, *497*, *502*, *507*
plot_trajectories(), *372*, *374*, *376*, *378*, *380*, *383*, *385*, *387*, *389*, *391*, *396*, *404*
plot_trajectories_fitted, 394, *394*
plot_trajectories_fitted(), *24*, *26*, *230*, *300*, *370*, *372*, *374*, *376*, *378*, *380*, *383*, *385*, *387*, *389*, *391*, *394*, *404*
polychoric, *467*, *468*
power_density_equivalence_md, 397, *402*, *404*
power_density_equivalence_md(), *233*, *235*, *238*, *376*, *400*, *402*, *404*, *696*
power_equivalence_c, *233*, 398, *694*, *696*
power_equivalence_c(), *233*, *235*, *238*, *376*, *398*, *402*, *404*, *696*
power_equivalence_md, *397–400*, 401, *404*
power_equivalence_md(), *233*, *235*, *238*, *376*, *398*, *400*, *404*, *696*
power_equivalence_md_plot, *398*, *402*, 403
power_equivalence_md_plot(), *233*, *235*, *238*, *372*, *374*, *376*, *378*, *380*, *383*, *385*, *387*, *389*, *391*, *394*, *396*, *398*, *400*, *402*, *696*
power_fisher_exact, 405
power_fisher_exact(), *574*, *659*, *661*, *664*, *670*, *674*, *678*, *682*, *685*, *690*, *693*, *696*, *698*, *700*, *702*, *705*, *707*, *710*, *713*, *716*, *719*, *721*, *725*, *730*, *732*, *733*, *736*, *738*, *741*
prcomp, *73*
predict, *13*
prime_time_achievement, 407
print_anova, *262*, *263*, 416, *418*
print_summary, *262*, *263*, *416*, 417
probability_of_superiority_paired, *36*, 418, *423*, *437*
probability_of_superiority_paired(), *135*, *137*, *176*, *240*, *243*, *246*, *248*, *252*, *256*, *356*, *362*, *365*, *423*, *471*, *518*
procrustes_phi, 420
procrustes_phi(), *28*, *40*, *46*, *49*, *55*, *73*, *148*, *150*, *218*, *227*, *274*, *281*, *285*, *304*, *310*, *480*
proportion_of_superiority, *419*, 422, *471*
proportion_of_superiority(), *135*, *137*, *176*, *240*, *243*, *246*, *248*, *252*, *256*, *356*, *362*, *365*, *420*, *471*, *518*
ptukey, *41–43*
pwr.f2.test, *692*
pygmalion, 423, *441*, *749*, *750*

qbryant_paulson, *63*, *189*, *190*, *752*
qbryant_paulson *(*bryant_paulson*)*, 41
qf, *197*
qtukey, *41*, *43*, *202*
quantile, *207*

R2_mixed_effects, 426, *429*, *430*, *761*
R2_mixed_effects(), *155*, *267*, *277*, *289*, *294*, *295*, *298*, *332*, *430*, *575*, *577*, *705*, *741*, *795*
R2_mixed_effects_decomposition, 429
R2_mixed_effects_decomposition(), *277*, *298*, *332*, *428*, *575*, *577*, *705*, *741*
randomization_test, *386*, *387*, 431
randomization_test(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *235*, *238*, *258*, *297*, *300*, *332*, *359*, *370*, *437*, *441*, *477*, *748*, *797*
randomization_test_paired, *36*, 435, *435*
randomization_test_paired(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *235*, *238*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *441*, *477*, *748*, *797*
regions_of_significance, *383*, *388*, *389*, 438, *772*
regions_of_significance(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *235*, *238*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *477*, *748*, *797*
reliability, *174*, *175*, *327*, *328*, 441, *451*, *454*, *456*, *464*, *468*, *614*, *617*, *762*, *767*, *768*
reliability(), *143*, *214*, *261*, *276*, *451*, *454*, *456*, *464*, *468*
reliability_alpha, *441*, *442*, *444*, 445, *453–456*, *464*, *615*, *619*, *762*, *767*, *768*
reliability_alpha(), *143*, *214*, *261*, *276*, *444*, *454*, *456*, *464*, *468*
reliability_H, *40*, *55*, *421*, 452

reliability_H(), *143*, *214*, *261*, *276*, *444*, *451*, *456*, *464*, *468*
reliability_kr20, *441*, *444*, *451*, 454, *468*, *762*
reliability_kr20(), *143*, *214*, *261*, *276*, *444*, *451*, *454*, *464*, *468*
reliability_omega, *28*, *40*, *46*, *55*, *175*, *274*, *285*, *310*, *441–444*, *446*, *447*, *451*, *454*, 457, *466*, *468*, *615*, *619*, *762*
reliability_omega(), *143*, *214*, *261*, *276*, *444*, *451*, *454*, *456*, *468*
reliability_omega_categorical, *46*, *52*, *281*, *441*, *444*, *446*, *448*, *451*, *455*, *456*, *460*, *464*, 465, *762*
reliability_omega_categorical(), *143*, *214*, *261*, *276*, *444*, *451*, *454*, *456*, *464*
responder_analysis, *98*, 469
responder_analysis(), *135*, *137*, *176*, *240*, *243*, *246*, *248*, *252*, *256*, *356*, *362*, *365*, *420*, *423*, *518*
results_sentence *(*dmar_output_helpers*)*, 218

sd, *472*
sd_unbiased, 471
sem, *182*, *338*, *341*, *342*, *532*, *622*, *623*, *634*, *635*, *637*, *638*, *688*
signal_to_noise_R2, 473
simple_effects_AB, 475
simple_effects_AB(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *235*, *238*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *441*, *748*, *797*
simple_structure, 478
simple_structure(), *28*, *40*, *46*, *49*, *55*, *73*, *148*, *150*, *218*, *227*, *274*, *281*, *285*, *304*, *310*, *421*
simulate_ancova_data, 480, *484*, *485*, *487*, *510*
simulate_ancova_data(), *485*, *487*, *490*, *493*, *497*, *502*, *507*, *510*
simulate_ancova_factorial_data, 483
simulate_ancova_factorial_data(), *482*, *487*, *490*, *493*, *497*, *502*, *507*, *510*
simulate_anova_data, *485*, 486, *510*
simulate_anova_data(), *482*, *485*, *490*, *493*, *497*, *502*, *507*, *510*
simulate_longitudinal_gompertz, *17*, 487, *493*, *497*, *507*
simulate_longitudinal_gompertz(), *482*, *485*, *487*, *493*, *497*, *502*, *507*, *510*
simulate_longitudinal_logistic, *17*, *490*, 491, *497*, *507*
simulate_longitudinal_logistic(), *482*, *485*, *487*, *490*, *497*, *502*, *507*, *510*
simulate_longitudinal_negative_exponential, *17*, *490*, *493*, 495, *507*
simulate_longitudinal_negative_exponential(), *482*, *485*, *487*, *490*, *493*, *502*, *507*, *510*
simulate_longitudinal_polynomial, *15*, *17*, *489*, *490*, *492*, *493*, *496*, *497*, 498, *506*, *507*
simulate_longitudinal_polynomial(), *482*, *485*, *487*, *490*, *493*, *497*, *507*, *510*
simulate_longitudinal_richards, *17*, *489*, *490*, *493*, *497*, 504
simulate_longitudinal_richards(), *482*, *485*, *487*, *490*, *493*, *497*, *502*, *510*
simulate_regression_data, *485*, *487*, 508
simulate_regression_data(), *482*, *485*, *487*, *490*, *493*, *497*, *502*, *507*
skewness, *290*, *291*, 510
skewness(), *207*, *291*
smd, *122*, *124*, *135*, *139*, *140*, *165*, *171*, *216*, *218*, *222*, *237*, *239*, *254–256*, *329*, *330*, *348*, *356*, *361*, *391*, *423*, *433*, *435*, 512, *515*, *518*, *649*, *738*, *748*, *789*, *793*, *797*
smd_c, *122*, *124*, 514, *514*, *649*
smd_trimmed, 516, *790*
smd_trimmed(), *135*, *137*, *176*, *240*, *243*, *246*, *248*, *252*, *256*, *356*, *362*, *365*, *420*, *423*, *471*
ss_aipe_c, *400*, 518, *521*, *525*, *526*, *625*, *633*, *659*, *694*, *696*, *744*, *746*
ss_aipe_c_ancova, *482*, *519*, 520, *661*
ss_aipe_c_ancova_sensitivity, 522
ss_aipe_c_sensitivity, 524
ss_aipe_c_sensitivity(), *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *565*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_cliff_delta, *137*, 526, *529*, *530*

ss_aipe_cliff_delta(), *526*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *565*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_cliff_delta_sensitivity, 528
ss_aipe_cliff_delta_sensitivity(), *526*, *528*, *534*, *553*, *555*, *558*, *560*, *563*, *565*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_composite_sem, *181*, 530, *569, 570*, *689, 690*
ss_aipe_composite_sem(), *526*, *528*, *530*, *553*, *555*, *558*, *560*, *563*, *565*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_crd, 535
ss_aipe_crd_both_fixed_budget *(*ss_aipe_crd*)*, 535
ss_aipe_crd_both_fixed_width *(*ss_aipe_crd*)*, 535
ss_aipe_crd_es, 540
ss_aipe_crd_es_both_fixed_budget *(*ss_aipe_crd_es*)*, 540
ss_aipe_crd_es_both_fixed_width *(*ss_aipe_crd_es*)*, 540
ss_aipe_crd_es_n_clusters_fixed_budget *(*ss_aipe_crd_es*)*, 540
ss_aipe_crd_es_n_clusters_fixed_width *(*ss_aipe_crd_es*)*, 540
ss_aipe_crd_es_n_individuals_fixed_budget *(*ss_aipe_crd_es*)*, 540
ss_aipe_crd_es_n_individuals_fixed_width *(*ss_aipe_crd_es*)*, 540
ss_aipe_crd_n_clusters_fixed_budget *(*ss_aipe_crd*)*, 535
ss_aipe_crd_n_clusters_fixed_width *(*ss_aipe_crd*)*, 535
ss_aipe_crd_n_individuals_fixed_budget *(*ss_aipe_crd*)*, 535
ss_aipe_crd_n_individuals_fixed_width *(*ss_aipe_crd*)*, 535
ss_aipe_cv, *185*, 547, *549, 550*, *769, 770*
ss_aipe_cv_sensitivity, 548, *548*, *770*
ss_aipe_equivalence_r, 551, *554, 555*
ss_aipe_equivalence_r(), *526*, *528*, *530*, *534*, *555*, *558*, *560*, *563*, *565*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_equivalence_r_sensitivity, *552*, 553, *553*
ss_aipe_equivalence_r_sensitivity(), *526*, *528*, *530*, *534*, *553*, *558*, *560*, *563*, *565*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_equivalence_smd, *237*, *553*, 556, *559, 560*
ss_aipe_equivalence_smd(), *526*, *528*, *530*, *534*, *553*, *555*, *560*, *563*, *566*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_equivalence_smd_sensitivity, *555*, *557*, 558
ss_aipe_equivalence_smd_sensitivity(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *563*, *566*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_icc, *212*, *277*, 560, *564, 565*, *574*, *795*
ss_aipe_icc(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *566*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_icc_sensitivity, *561, 562*, 563
ss_aipe_icc_sensitivity(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_indirect_effect, *312*, *314*, *322*, 566, *571, 572*, *701, 702*, *776, 777*
ss_aipe_indirect_effect(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *572*, *577*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_indirect_effect_sensitivity, *569, 570*, 571
ss_aipe_indirect_effect_sensitivity(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *570*, *577*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_mixed_effects, 573, *576, 577*
ss_aipe_mixed_effects(), *277*, *298*, *332*, *407*, *428*, *430*, *577*, *659*, *661*, *664*, *670*, *674*, *678*, *682*, *685*, *690*, *693*,

696, *698*, *700*, *702*, *705*, *707*, *710*, *713*, *716*, *719*, *721*, *725*, *730*, *732*, *733*, *736*, *738*, *741*
ss_aipe_mixed_effects_sensitivity, 575, *591*
ss_aipe_mixed_effects_sensitivity(), *277*, *298*, *332*, *428*, *430*, *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *570*, *572*, *575*, *579*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*, *705*, *741*
ss_aipe_omega_squared, 577, *580, 581*, *779*
ss_aipe_omega_squared(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *570*, *572*, *577*, *581*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_omega_squared_sensitivity, 580
ss_aipe_omega_squared_sensitivity(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *570*, *572*, *577*, *579*, *584*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_partial_r, *248*, *528*, *570*, 582, *585, 586*, *593, 594*, *640, 641*
ss_aipe_partial_r(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *570*, *572*, *577*, *579*, *581*, *586*, *591*, *594*, *596*, *620*, *641*, *643*
ss_aipe_partial_r_sensitivity, *583*, 585, *596*, *643*
ss_aipe_partial_r_sensitivity(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *570*, *572*, *577*, *579*, *581*, *584*, *592*, *594*, *596*, *620*, *641*, *643*
ss_aipe_pcm, 587, *590, 591*
ss_aipe_pcm_sensitivity, *588*, 590
ss_aipe_pcm_sensitivity(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *594*, *596*, *620*, *641*, *643*
ss_aipe_r, *67*, *553*, *583, 584*, 592, *594–596*
ss_aipe_r(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *592*, *596*, *620*, *641*, *643*
ss_aipe_R2, *67*, *92*, *103*, *170*, *254*, *474*, *510*, *578, 579*, 596, *602*, *641*, *716*, *766*, *785*
ss_aipe_R2_sensitivity, *598*, 599, *717*, *719*
ss_aipe_r_sensitivity, *555*, *593*, 594, *594*
ss_aipe_r_sensitivity(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *592*, *594*, *620*, *641*, *643*
ss_aipe_rc, *570*, 602, *654*
ss_aipe_rc_sensitivity, 604, *657*
ss_aipe_reg_coef, *106*, *109*, *129*, *510*, *604*, 607, *607*, *613*, *654*, *657*, *721*, *725*
ss_aipe_reg_coef_sensitivity, *604*, *606, 607*, *610*, 611, *654*, *656, 657*, *727*, *729*
ss_aipe_reliability, *184*, *444*, *451*, 614, *618, 619*, *768*, *775*
ss_aipe_reliability_sensitivity, 618
ss_aipe_reliability_sensitivity(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *592*, *594*, *596*, *641*, *643*
ss_aipe_rmsea, 620, *623*
ss_aipe_rmsea_sensitivity, *181, 182*, 621
ss_aipe_sc, *519*, 624, *628*, *633*, *733*
ss_aipe_sc_ancova, *20*, 626, *631*
ss_aipe_sc_ancova_sensitivity, *627*, 628, *628*
ss_aipe_sc_sensitivity, *526*, 631, *631*
ss_aipe_sem_path, *181, 182*, *532–534*, 633, *637, 638*, *690*
ss_aipe_sem_path_sensitivity, *181, 182*, *534*, *634, 635*, 636
ss_aipe_semipartial_r, *528*, *570*, *584*, 639, *642, 643*
ss_aipe_semipartial_r(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *592*, *594*, *596*, *620*, *643*
ss_aipe_semipartial_r_sensitivity, *586*, *640*, 641
ss_aipe_semipartial_r_sensitivity(), *526*, *528*, *530*, *534*, *553*, *555*, *558*, *560*, *563*, *566*, *570*, *572*, *577*, *579*, *581*, *584*, *586*, *592*, *594*, *596*, *620*, *641*
ss_aipe_sm, 643, *647*
ss_aipe_sm_sensitivity, 645
ss_aipe_smd, *122*, *171*, *210*, *222*, *256*, *514*,

557, 647, *651*, *737, 738*, *788–790*
ss_aipe_smd_sensitivity, *560*, 649
ss_aipe_src, *604*, 652
ss_aipe_src_sensitivity, *607*, 655
ss_power_c, 657, *661*, *670*, *700*, *707*, *733*, *759*
ss_power_c(), *407*, *574*, *661*, *664*, *670*, *674*, *678*, *682*, *685*, *690*, *693*, *696*, *698*, *700*, *702*, *705*, *707*, *710*, *713*, *716*, *719*, *721*, *725*, *730*, *732, 733*, *736*, *738*, *741*
ss_power_c_ancova, 659, *659*, *667–670*, *759*
ss_power_c_ancova(), *407*, *574*, *659*, *664*, *670*, *674*, *678*, *682*, *685*, *690*, *693*, *696*, *698*, *700*, *702*, *705*, *707*, *710*, *713*, *716*, *719*, *721*, *725*, *730*, *732, 733*, *736*, *738*, *741*
ss_power_composite_ancova, 661, *667*, *670*, *673, 674*
ss_power_composite_ancova(), *407*, *574*, *659*, *661*, *670*, *674*, *678*, *682, 683*, *685*, *690*, *693*, *696*, *698*, *700*, *702*, *705*, *707*, *710*, *713*, *716*, *719*, *721*, *725*, *730*, *732, 733*, *736*, *738*, *741*
ss_power_composite_ancova_2group, *663, 664*, 665, *678, 679*, *681, 682*
ss_power_composite_ancova_2group(), *407*, *574*, *659*, *661*, *664*, *674*, *678*, *682, 683*, *685*, *690*, *693*, *696*, *698*, *700*, *703*, *705*, *707*, *710*, *713*, *716*, *719*, *721*, *725*, *730*, *732*, *734*, *736*, *738*, *741*
ss_power_composite_anova, *663, 664*, 672, *690*
ss_power_composite_anova(), *407*, *575*, *659*, *661*, *664*, *670*, *678*, *682, 683*, *685*, *690*, *693*, *696*, *698*, *700*, *703*, *705*, *707*, *710*, *713*, *716*, *719*, *721*, *725*, *730*, *732*, *734*, *736*, *738*, *741*
ss_power_composite_factorial_ancova, *663, 664*, 675, *681, 682*, *684, 685*
ss_power_composite_factorial_ancova(), *407*, *575*, *659*, *661*, *664*, *670*, *674*, *682, 683*, *685*, *690*, *693*, *696*, *698*, *700*, *703*, *705*, *707*, *710*, *713*, *716*, *719*, *721*, *725*, *730*, *732*, *734*, *736*, *738*, *741*
ss_power_composite_factorial_ancova_het, *663, 664*, 679
ss_power_composite_factorial_ancova_het(), *407*, *575*, *659*, *661*, *664*, *670*, *674*, *678*, *685*, *690*, *693*, *696*, *698*, *700*, *703*, *705*, *707*, *710*, *713*, *716*, *719*, *722*, *726*, *730*, *732*, *734*, *736*, *738*, *741*
ss_power_composite_factorial_anova, *676*, *678*, 683
ss_power_composite_factorial_anova(), *407*, *575*, *659*, *661*, *664*, *670*, *674*, *678*, *682, 683*, *690*, *693*, *696*, *698*, *700*, *703*, *705*, *707*, *710*, *713*, *716*, *719*, *722*, *726*, *730*, *732*, *734*, *736*, *738*, *741*
ss_power_composite_sem, *181*, *531*, *534*, 686
ss_power_composite_sem(), *407*, *575*, *659*, *661*, *664*, *670*, *674*, *678*, *682, 683*, *685*, *693*, *696*, *698*, *700*, *703*, *705*, *707*, *710*, *713*, *716*, *719*, *722*, *726*, *730*, *732*, *734*, *736*, *738*, *741*
ss_power_contrast, *487*, 691, *695, 696*, *706*, *712*, *759*
ss_power_contrast(), *407*, *575*, *659*, *661*, *664*, *670*, *674*, *678*, *682*, *685*, *690*, *696*, *698*, *700*, *703*, *705*, *707*, *710*, *713*, *716*, *719*, *722*, *726*, *730*, *732*, *734*, *736*, *738*, *741*
ss_power_equivalence_c, *233*, *400*, 694, *744*, *746*, *759*
ss_power_equivalence_c(), *233*, *235*, *238*, *376*, *398*, *400*, *402*, *404*, *407*, *575*, *659*, *661*, *664*, *670*, *674*, *678*, *682*, *685*, *690*, *693*, *698*, *700*, *703*, *705*, *707*, *710*, *713*, *716*, *719*, *722*, *726*, *730*, *732*, *734*, *736*, *738*, *741*
ss_power_factorial_ancova, *664*, *676–678*, 696, *700*, *759*
ss_power_factorial_ancova(), *407*, *575*, *659*, *661*, *664*, *670*, *674*, *678*, *682*, *685*, *690*, *693*, *696*, *700*, *703*, *705*, *707*, *710*, *713*, *716*, *719*, *722*, *726*, *730*, *732*, *734*, *736*, *738*, *741*
ss_power_factorial_anova, *477*, *674*, *677, 678*, *685*, *696–698*, 699, *707*, *741*, *759*
ss_power_factorial_anova(), *407*, *575*, *659*, *661*, *664*, *670*, *674*, *678*, *682*,

685, 690, 693, 696, 698, 703, 705, 707, 710, 713, 716, 719, 722, 726, 730, 732, 734, 736, 738, 741
ss_power_indirect_effect, 312, 314, 322, 701, 759
ss_power_indirect_effect(), 314, 322, 407, 575, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 705, 707, 710, 713, 716, 719, 722, 726, 730, 732, 734, 736, 738, 741
ss_power_mixed_effects, 428, 574, 703, 740, 741, 759
ss_power_mixed_effects(), 277, 298, 332, 407, 428, 430, 575, 577, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 703, 707, 710, 713, 716, 719, 722, 726, 730, 732, 734, 736, 738, 741
ss_power_one_way_anova, 659, 674, 700, 705, 732, 733, 741, 759
ss_power_one_way_anova(), 407, 575, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 703, 705, 710, 713, 716, 719, 722, 726, 730, 732, 734, 736, 738, 741
ss_power_pcm, 498, 500–502, 588, 705, 707, 732
ss_power_pcm(), 407, 575, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 703, 705, 707, 713, 716, 719, 722, 726, 730, 732, 734, 736, 738, 741
ss_power_r, 67, 594, 711, 759
ss_power_r(), 407, 575, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 703, 705, 707, 710, 716, 719, 722, 726, 730, 732, 734, 736, 738, 741
ss_power_R2, 138, 713, 718, 719, 721, 725, 759
ss_power_R2(), 407, 574, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 702, 705, 707, 710, 713, 719, 721, 725, 730, 732, 733, 736, 738, 741
ss_power_R2_sensitivity, 716, 716, 759
ss_power_R2_sensitivity(), 407, 574, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 702, 705, 707, 710, 713, 716, 721, 725, 730, 732, 733, 736, 738, 741
ss_power_rc, 719
ss_power_rc(), 407, 575, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 703, 705, 707, 710, 713, 716, 719, 726, 730, 732, 734, 736, 738, 741
ss_power_reg_coef, 667, 668, 670, 693, 716, 721, 723, 729, 759
ss_power_reg_coef(), 407, 575, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 703, 705, 707, 710, 713, 716, 719, 722, 730, 732, 734, 736, 738, 741
ss_power_reg_coef_sensitivity, 727, 759
ss_power_reg_coef_sensitivity(), 407, 575, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 703, 705, 707, 710, 713, 716, 719, 722, 726, 732, 734, 736, 738, 741
ss_power_rm_anova, 730, 741, 759
ss_power_rm_anova(), 407, 575, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 703, 705, 707, 710, 713, 716, 719, 722, 726, 730, 734, 736, 738, 741
ss_power_sc, 659, 707, 732, 759
ss_power_sc(), 407, 575, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 703, 705, 707, 710, 713, 716, 719, 722, 726, 730, 732, 736, 738, 741
ss_power_sem, 688, 690, 734, 759
ss_power_sem(), 407, 575, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 703, 705, 707, 710, 713, 716, 719, 722, 726, 730, 732, 734, 738, 741
ss_power_smd, 122, 210, 222, 514, 647, 670, 705, 736, 759
ss_power_smd(), 407, 575, 659, 661, 664, 670, 674, 678, 682, 685, 690, 693, 696, 698, 700, 703, 705, 707, 710, 713, 716, 719, 722, 726, 730, 732, 734, 736, 741

ss_power_split_plot_anova, *705*, 739, *759*
ss_power_split_plot_anova(), *277*, *298*, *332*, *407*, *428*, *430*, *575*, *577*, *659*, *661*, *664*, *670*, *674*, *678*, *682*, *685*, *690*, *693*, *696*, *698*, *700*, *703*, *705*, *707*, *710*, *713*, *716*, *719*, *722*, *726*, *730*, *732*, *734*, *736*, *738*
ss_seq_c, 741, *744–746*
ss_seq_c(), *746*
ss_seq_c_sensitivity, *743*, 744, *744*
ss_seq_c_sensitivity(), *744*
stats::pchisq(), *90*
stats::pf(), *92*
stats::power.t.test(), *649*
stats::pt(), *94*
stats::qchisq(), *90*
stats::qf(), *92*
stats::qt(), *94*
summary.lm, *765*
summary_t_test, 746, *748*, *797*
summary_t_test(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *235*, *238*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *441*, *477*, *797*
summary_t_test_broom, 748
Sys.time, *357*

t.test, *83*, *432*, *435*, *437*, *748*, *797*
teacher_expectancy, *147*, *330*, *378*, 749
test_market, 751
tidy, *84*, *160*, *658*, *660*, *663*, *668*, *673*, *677*, *681*, *685*, *689*, *692*, *695*, *698*, *700*, *702*, *704*, *706*, *718*, *721*, *729*, *731*, *733*, *740*, *756*
tidy.dmar_cfa_k, 753
tidy.dmar_ci_anova *(*tidy.dmar_contrast_test*)*, 756
tidy.dmar_ci_long *(*tidy.dmar_contrast_test*)*, 756
tidy.dmar_ci_R2, 754
tidy.dmar_ci_smd, 755
tidy.dmar_content_validity *(*tidy.dmar_contrast_test*)*, 756
tidy.dmar_contrast_test, 756
tidy.dmar_dmacs *(*tidy.dmar_contrast_test*)*, 756
tidy.dmar_measurement_alignment *(*tidy.dmar_contrast_test*)*, 756
tidy.dmar_measurement_invariance *(*tidy.dmar_contrast_test*)*, 756
tidy.dmar_mediation_mbco, 760
tidy.dmar_post_hoc_ci *(*tidy.dmar_contrast_test*)*, 756
tidy.dmar_R2_mixed_effects, 761
tidy.dmar_reliability, 762
tidy.dmar_ss_aipe *(*tidy.dmar_contrast_test*)*, 756
tidy.dmar_ss_power *(*tidy.dmar_contrast_test*)*, 756
tidy.dmar_ss_power_sensitivity *(*tidy.dmar_contrast_test*)*, 756
tidy.dmar_summary_t_test *(*summary_t_test_broom*)*, 748
tidy.dmar_tbl *(*tidy.dmar_contrast_test*)*, 756
tidy.dmar_welch_t *(*welch_t_broom*)*, 798
tidy.mlmr, 763
tidy.mlmr_mv, 764
TukeyHSD, *132*, *160*, *202*

unbiased_R2, 765
uniroot, *42*, *88–94*, *102*, *193*, *199*, *501*

var.test, *359*
var_alpha, 767
var_alpha(), *770*, *772*, *777*, *779*, *783*, *789*, *790*
var_cv, *185*, 769
var_cv(), *768*, *772*, *777*, *779*, *783*, *789*, *790*
var_ete, 770
var_ete(), *768*, *770*, *777*, *779*, *783*, *789*, *790*
var_icc, *212*, *276*, *277*, *562*, *565*, *574*, 773, *795*
var_indirect_effect, *313*, *314*, *322*, *572*, *702*, 775
var_indirect_effect(), *768*, *770*, *772*, *779*, *783*, *789*, *790*
var_omega_squared, 777
var_omega_squared(), *768*, *770*, *772*, *777*, *783*, *789*, *790*
var_partial_r, *248*, *584*, 780, *783*, *787*
var_r, *67*, 782
var_r(), *768*, *770*, *772*, *777*, *779*, *789*, *790*
var_R2, *254*, *766*, *775*, *781*, 784, *787*
var_semipartial_r, *640*, *641*, *781*, *783*, 785
var_smd, *348*, 787, *790*
var_smd(), *768*, *770*, *772*, *777*, *779*, *783*, *790*

var_smd_trimmed, *518*, 789
var_smd_trimmed(), *768*, *770*, *772*, *777*, *779*, *783*, *789*
VarCorr, *427, 428*
vargha_delaney_A, 791
variance_components_mls, *276, 277*, 794
variance_components_mls(), *155*, *267*, *277*, *289*, *294, 295*, *428*
vcov, *342*

welch_t, *31*, *747, 748*, 796, *799*
welch_t(), *14*, *20*, *24*, *71*, *117*, *132*, *152*, *160*, *179*, *235*, *238*, *258*, *297*, *300*, *332*, *359*, *370*, *435*, *437*, *441*, *477*, *748*
welch_t_broom, 798
wilcox.test, *435*
writeLines, *178*