Package ‘MBESS’

June 6, 2017

Type Package

Title The MBESS R Package

Version 4.3.0

Date 2017-07-05

Maintainer Ken Kelley <kkelley@nd.edu>

Depends R (>= 3.2.0), stats

Imports boot, gsl, lavaan, MASS, methods, mnormt, nlme, OpenMx, parallel, sem, semTools

Description Implements methods that useful in designing research studies and analyzing data, with particular emphasis on methods that are developed for or used within the behavioral, educational, and social sciences (broadly defined). That being said, many of the methods implemented within MBESS are applicable to a wide variety of disciplines. MBESS has a suite of functions for a variety of related topics, such as effect sizes, confidence intervals for effect sizes (including standardized effect sizes and noncentral effect sizes), sample size planning (from the accuracy in parameter estimation [AIPE], power analytic, equivalence, and minimum-risk point estimation perspectives), mediation analysis, various properties of distributions, and a variety of utility functions. MBESS (pronounced 'em-bes') was originally an acronym for 'Methods for the Behavioral, Educational, and Social Sciences,' but at this point MBESS contains methods applicable and used in a wide variety of fields and is an orphan acronym, in the sense that what was an acronym is now literally its name. MBESS has greatly benefited from others, see <http://nd.edu/~kkelley/site/MBESS.html> for a detailed list of those that have contributed and other details.

License GPL-2 | GPL-3

URL http://nd.edu/~kkelley/site/MBESS.html

NeedsCompilation no

Author Ken Kelley [aut, cre]

Repository CRAN

Date/Publication 2017-06-06 16:00:39 UTC

R topics documented:

`aipe.smd` *Sample size planning for the standardized mean different from the accuracy in parameter estimation approach*

Description

A set of functions that `ss.aipe.smd` calls upon to calculate the appropriate sample size for the standardized mean difference such that the expected value of the confidence interval is sufficiently narrow.

Usage

```
ss.aipe.smd.full(delta, conf.level, width, ...)
ss.aipe.smd.lower(delta, conf.level, width, ...)
ss.aipe.smd.upper(delta, conf.level, width, ...)
```

Arguments

`delta`	the population value of the standardized mean difference
`conf.level`	the desired degree of confidence (i.e., 1-Type I error rate)
`width`	desired width of the specified (i.e., `Lower`, `Upper`, `Full`) region of the confidence interval
`...`	specify additional parameters in functions these functions call upon

Value

`n`	The necessary sample size *per group* in order to satisfy the specified goals.

Warning

The returned value is the sample size *per group*. Currently only `ss.aipe.smd.full` returns the exact value. However, `ss.aipe.smd.lower` and `ss.aipe.smd.upper` provide approximate sample size values.

Note

The function `ss.aipe.smd` is the function users should generally use. The function `ss.aipe.smd` calls upon these functions as needed. They can be thought of loosely as internal MBESS functions.

Author(s)

Ken Kelley (University of Notre Dame; <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*, 532–574.

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 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.

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*, 258–287.

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.

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`

`ancova.random.data` *Generate random data for an ANCOVA model*

Description

Generate random data for a simple (one-response-one-covariate) ANCOVA model considering the covariate as random. Data can be generated in the contexts of both randomized design (same population covariate mean across groups) and non-randomized design (different population covariate means across groups).

Usage

```
ancova.random.data(mu.y, mu.x, sigma.y, sigma.x, rho, J, n, randomized = TRUE)
```

Arguments

`mu.y`	a vector of the population group means of the response variable
`mu.x`	the population mean of the covariate (in the randomized design context), or a vector of the population group means of the covariate (in the non-randomized design context)

`sigma.y`	the population standard deviation of the response (outcome) variable
`sigma.x`	the population standard deviation of the covariate
`rho`	the population correlation coefficient between the response and the covariate
`J`	the number of groups
`n`	the number of sample size *per group*
`randomized`	a logical statement of whether randomized design is used

Details

This function uses a multivariate normal distribution to generate the random data; the covariate is considered as a random variable in the model. This function uses `mvrnorm` in the `MASS` package in an internal function, and thus it requires the `MASS` package be installed.

This function assumes homogeneous covariance matrix among groups, in both the randomized design and non-randomized design contexts.

Value

This function returns an n by $J2$ matrix, where n and J are as defined in the argument. The first J columns of the matrix contains the random data for the response, and the second J columns of the matrix contains the random data for the covariate.

Author(s)

Keke Lai (University of California-Merced) and Ken Kelley (University of Notre Dame) <kkelley@nd.edu>

See Also

`mvrnorm` in the `MASS` package

Examples

```
random.data <- ancova.random.data(mu.y=c(3,5), mu.x=10, sigma.y=1,
sigma.x=2, rho=.8, J=2, n=20)
```

`CFA.1` *One-factor confirmatory factor analysis model*

Description

Returns the MLE estimates and the estimated asymptotic covariance matrix of parameter estimates for one-factor confirmatory factor analysis model

Usage

```
CFA.1(S, N, equal.loading = FALSE, equal.error = FALSE, package="lavaan",
se="standard", ...)
```

Arguments

`S`	covariance matrix of the indicators
`N`	total sample size
`equal.loading`	logical statement indicating whether the path coefficients are the same
`equal.error`	logical statement indicating whether the manifest variables have the same error variances
`package`	the package used in confirmatory factor analysis (`sem` or `lavaan`
`se`	See the `cfa` and check the `se` argument
`...`	Additional arguments for the `cfa` function

Value

`Model`	the factor analysis model specified by the user
`Factor.Loadings`	factor loadings
`Indicator.var`	the error variances of the indicator variables
`Parameter.cov`	the covariance matrix of the parameters
`converged`	`TRUE` or `FALSE` statement on if the model converged
`package`	notes the package used to get the output

Note

The output will differ slightly, both in form and potentially values, based on which package **lavaan** or **sem** is used.

Author(s)

Keke Lai (University of California-Merced) and Ken Kelley (University of Notre Dame)

See Also

`sem`, `covmat.from.cfm`

Examples

```
## Not run:
cov.mat<- matrix(
c(1.384, 1.484, 1.988, 2.429, 3.031,
1.484, 2.756, 2.874, 3.588, 4.390,
1.988, 2.874, 4.845, 4.894, 6.080,
2.429, 3.588, 4.894, 6.951, 7.476,
3.031, 4.390, 6.080, 7.476, 10.313), nrow=5)


CFA.1(N=300, S=cov.mat, package="lavaan")

CFA.1(N=300, S=cov.mat, package="sem")
```

```
## End(Not run)
```

---

ci.c *Confidence interval for a contrast in a fixed effects ANOVA*

Description

Function to calculate the exact confidence interval for a contrast in a fixed effects analysis of variance context. This function assumes homogeneity of variance (as does the ANOVA upon 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 (unstandardized) contrast effect, obtained by multiplying the *j*th mean by the jth contrast weight (this is the unstandardized effect)
`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

Returns the confidence limits for the contrast:

`Lower.Conf.Limit.Contrast`
: the lower confidence limit for the contrast effect

`Contrast`
: the value of the estimated unstandardized contrast effect

`Upper.Conf.Limit.Contrast`
: the upper confidence limit for the contrast effect

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 is zero.

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>)

References

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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*, 164–182.

See Also

`conf.limits.nct`, `ci.sc`, `ci.src`, `ci.smd`, `ci.smd.c`, `ci.sm`

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 and Delaney (2004, pp. 155--171) :
# 24 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

To calculate the confidence interval for an unstandardized contrast in the one-covariate ANCOVA.

Usage

```
ci.c.ancova(Psi, adj.means, s.ancova = NULL, c.weights, n,
cov.means, SSwithin.x, conf.level = 0.95, ...)
```

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)
`...`	allows one to potentially include parameter values for inner functions

Value

`lower.limit`	the lower confidence limit of the (unstandardized) ANCOVA contrast
`upper.limit`	the upper confidence limit of the (unstandardized) ANCOVA contrast

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)

Keke Lai (University of California–Merced) and Ken Kelley (University of Notre Dame; `<kkelley@nd.edu>`)

References

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

Maxwell, S. E., & Delaney, H. D. (2004). *Designing experiments and analyzing data: A model comparison perspective*. Mahwah, NJ: Erlbaum.

See Also

`ci.c`, `ci.sc.ancova`

Examples

```
# Maxwell & Delaney (2004, pp. 428-468) 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
# 313.37.

# 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=313.37)
```

ci.cc *Confidence interval for the population correlation coefficient*

Description

This function is used to form a confidence interval for the population correlation coefficient. Note that this appraoch assumes that the variables the sample correlation coefficient are based are assumed to be bivariate normally distributed (e.g., Hays, 1994, Chapter 14).

Usage

```
ci.cc(r, n, conf.level = 0.95, alpha.lower = NULL, alpha.upper = NULL)
```

Arguments

`r`	observed value of the correlation coefficient (specifically the zero-order Pearson product-moment correlation coefficient)
`n`	sample size
`conf.level`	desired confidence level, where the error rate is the same on each side
`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

Details

Note that this appraoch to confidence intervals does will not generally lead to a symmetric confidence interval. The function first transforms r into emphZ\' , forms a confidence interval for the population value (i.e., $zeta$), and then transforms the confidence limits for $zeta$ into the scale of the correlation coefficient.

Value

`Lower.Limit`	lower limit of the confidence interval
`Estimated.Correlation`	observed value of the correlation coefficient
`Upper.Limit`	upper limit of the confidence interval

Note

This confidence interval assumes that the two variables the correlation is based are bivariate normal. See Hays (2004, Chapter 14) for details.

Author(s)

Ken Kelley (University of Notre Dame; `<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.

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

See Also

`transform_Z.r`, `transform_r.Z`

Examples

```
# Example, from Hayes. Suppose n=100 and r=.35.
ci.cc(r=.35, n=100, conf.level=.95)

# Here is another way to enter the above example.
ci.cc(r=.35, n=100, conf.level=NULL, alpha.lower=.025, alpha.upper=.025)

# Here are examples of one-sided confidence intervals.
ci.cc(r=.35, n=100, conf.level=NULL, alpha.lower=0, alpha.upper=.05)
ci.cc(r=.35, n=100, conf.level=NULL, alpha.lower=.05, alpha.upper=0)
```

`ci.cv` *Confidence interval for the coefficient of variation*

Description

Function to calculate the confidence interval for the population coefficient of variation using the noncentral `t`-distribution.

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

`Lower.Limit.CofV`
: Lower confidence interval limit

`Prob.Less.Lower`
: Proportion of the distribution beyond `Lower.Limit.CofV`

`Upper.Limit.CofV`
: Upper confidence interval limit

`Prob.Greater.Upper`
: Proportion of the distribution beyond `Upper.Limit.CofV`

`C.of.V` Observed coefficient of variation

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`)

References

Johnson, B. L., & Welch, B. L. (1940). Applications of the non-central t-distribution. *Biometrika*, 31, 362–389.

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.

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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

See Also

`cv`

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.pvaf	*Confidence Interval for the Proportion of Variance Accounted for (in the dependent variable by knowing the levels of the factor)*

Description

Function to obtain the exact confidence limits for the proportion of variance of the dependent variable accounted for by knowing the levels of the factor (or the grouping factor in a single factor design) group status in a fixed factor analysis of variance.

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 `conf.limits.ncf` in MBESS, which is used within this function.

Value

Returns the confidence interval for the proportion of variance of the dependent variable accounted for by knowing group status in a fixed factor analysis of variance (using a noncentral *F*-distribution).

`Lower.Limit.Proportion.of.Variance.Accounted.for`
: The lower confidence limit for the proportion of variance accounted for in the deviation by group status.

`Upper.Limit.Proportion.of.Variance.Accounted.for`
: The upper confidence limit for the proportion of variance accounted for in the deviation by group status.

Note

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

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`)

References

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

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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*, 164–182.

See Also

`conf.limits.ncf`

Examples

```
## Not run:
## 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 11.2213. 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)

## End(Not run)
```

ci.R *Confidence interval for the multiple correlation coefficient*

Description

A function to obtain the confidence interval for the population multiple correlation coefficient when predictors are random (the default) or fixed.

Usage

```
ci.R(R = NULL, df.1 = NULL, df.2 = NULL, conf.level = 0.95,
Random.Predictors = TRUE, Random.Regressors, F.value = NULL,
N = NULL, K=NULL, alpha.lower = NULL, alpha.upper = NULL, ...)
```

Arguments

`R`	multiple correlation coefficient
`df.1`	numerator degrees of freedom
`df.2`	denominator degrees of freedom
`conf.level`	confidence interval coverage (i.e., 1- Type I error rate); default is .95
`Random.Predictors`	whether or not the predictor variables are random or fixed (random is default)
`Random.Regressors`	an alias for `Random.Predictors`; `Random.Regressors` overrides `Random.Predictors`
`F.value`	obtained F-value
`N`	sample size
`K`	number of predictors
`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

This function is based on the function `ci.R2` in MBESS package.

This function can be used with random predictor variables (`Random.Predictors=TRUE`) or when predictor variables are fixed (`Random.Predictors=FALSE`). In many applications in the behavioral, educational, and social sciences, predictor variables are random, which is the default for this function.

For random predictors, the function implements the procedure of Lee (1971), which was implemented by Algina and Olejnik (2000; specifically in their *ci.smcc.bisec.sas* SAS script). When `Random.Predictors=TRUE`, the function implements code that is in part based on the Alginia and Olejnik (2000) SAS script.

When `Random.Predictors=FALSE`, and thus the predictors are planned and thus fixed in hypothetical replications of the study, the confidence limits are based on a noncentral F-distribution (see `conf.limits.ncf`).

Value

`Lower.Conf.Limit.R`
: lower limit of the confidence interval around the population multiple correlation coefficient

`Prob.Less.Lower`
: proportion of the distribution less than `Lower.Conf.Limit.R`

`Upper.Conf.Limit.R`
: upper limit of the confidence interval around the population multiple correlation coefficient

`Prob.Greater.Upper`
: proportion of the distribution greater than `Upper.Conf.Limit.R`

Author(s)

Ken Kelley (University of Notre Dame; <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*, 119–136.

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

Smithson, M. (2003). *Confidence intervals*. New York, NY: 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*, 164–182.

Steiger, J. H. & Fouladi, R. T. (1992). R2: A computer program for interval estimation, power calculation, and hypothesis testing for the squared multiple correlation. *Behavior research methods, instruments and computers, 4*, 581–582.

See Also

`ci.R2`, `ss.aipe.R2`, `conf.limits.nct`

`ci.R2` — *Confidence interval for the population squared multiple correlation coefficient*

Description

A function to calculate the confidence interval for the population squared multiple correlation coefficient.

Usage

```
ci.R2(R2 = NULL, df.1 = NULL, df.2 = NULL, conf.level = .95,
Random.Predictors=TRUE, Random.Regressors, F.value = NULL, N = NULL,
p = NULL, K, alpha.lower = NULL, alpha.upper = NULL, tol = 1e-09)
```

Arguments

R2	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)
Random.Regressors	an alias for Random.Predictors; Random.Regressors overrides Random.Predictors
F.value	obtained F-value
N	sample size
p	number of predictors
K	alias for p, the number of predictors
alpha.lower	Type I error for the lower confidence limit
alpha.upper	Type I error for the upper confidence limit
tol	tolerance for iterative convergence

Details

This function can be used with random predictor variables (Random.Predictors=TRUE) or when predictor variables are fixed (Random.Predictors=FALSE). In many applications of multiple regression, predictor variables are random, which is the default in this function.

For random predictors, the function implements the procedure of Lee (1971), which was implemented by Algina and Olejnik (2000; specifically in their *ci.smcc.bisec.sas* SAS script). When Random.Predictors=TRUE, the function implements code that is in part based on the Alginia and Olejnik (2000) SAS script.

When Random.Predictors=FALSE, and thus the predictors are planned and thus fixed in hypothetical replications of the study, the confidence limits are based on a noncentral F-distribution (see conf.limits.ncf).

Value

Lower.Conf.Limit.R2	upper limit of the confidence interval around the population multiple correlation coefficient
Prob.Less.Lower	proportion of the distribution less than Lower.Conf.Limit.R2
Upper.Conf.Limit.R2	upper limit of the confidence interval around the population multiple correlation coefficient
Prob.Greater.Upper	proportion of the distribution greater than Upper.Conf.Limit.R2

Author(s)

Ken Kelley (University of Notre Dame; <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*, 119–136.

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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

Smithson, M. (2003). *Confidence intervals*. New York, NY: Sage Publications.

Steiger, J. H. & Fouladi, R. T. (1992) R2: A computer program for interval estimation, power calculation, and hypothesis testing for the squared multiple correlation. *Behavior research methods, instruments and computers, 4*, 581–582.

See Also

`ss.aipe.R2`, `conf.limits.ncf`

Examples

```
# For random predictor variables.
# ci.R2(R2=.25, N=100, K=5, conf.level=.95, Random.Predictors=TRUE)

# ci.R2(F.value=6.266667, N=100, K=5, conf.level=.95, Random.Predictors=TRUE)

# For fixed predictor variables.
# ci.R2(R2=.25, N=100, K=5, conf.level=.95, Random.Predictors=TRUE)

# ci.R2(F.value=6.266667, N=100, K=5, conf.level=.95, Random.Predictors=TRUE)

# One sided confidence intervals when predictors are random.
# ci.R2(R2=.25, N=100, K=5, alpha.lower=.05, alpha.upper=0, conf.level=NULL,
# Random.Predictors=TRUE)

# ci.R2(R2=.25, N=100, K=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, K=5, alpha.lower=.05, alpha.upper=0, conf.level=NULL,
# Random.Predictors=FALSE)

# ci.R2(R2=.25, N=100, K=5, alpha.lower=0, alpha.upper=.05, conf.level=NULL,
# Random.Predictors=FALSE)
```

ci.rc *Confidence Interval for a Regression Coefficient*

Description

A function to calculate a confidence interval for the population regression coefficient of interest using the standard approach and the noncentral approach when the regression coefficients are standardized.

Usage

```
ci.rc(b.k, SE.b.k = NULL, s.Y = NULL, s.X = NULL, N, K, R2.Y_X = NULL,
R2.k_X.without.k = NULL, conf.level = 0.95, R2.Y_X.without.k = NULL,
t.value = NULL, alpha.lower = NULL, alpha.upper = NULL,
Noncentral = FALSE, Suppress.Statement = FALSE, ...)
```

Arguments

- `b.k` value of the regression coefficient for the *k*th predictor variable
- `SE.b.k` standard error for the *k*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
- `K` the number of predictors
- `R2.Y_X` the squared multiple correlation coefficient predicting *Y* from the *k* predictor variables
- `R2.k_X.without.k` the squared multiple correlation coefficient predicting the *k*th predictor variable (i.e., the predictor of interest) from the remaining *K*-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.k` the squared multiple correlation coefficient predicting *Y* from the *K*-1 predictor variable with the *k*th predictor of interest excluded
- `t.value` the *t*-value evaluating the null hypothesis that the population regression coefficient for the *k*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
- `Suppress.Statement` `TRUE` or `FALSE` statement specifying whether or not a statement should be printed that identifies the type of confidence interval formed
- `...` optional additional specifications for nested functions

Details

This function calls upon `ci.reg.coef` in MBESS, but has a different naming system. See `ci.reg.coef` for more details.

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

Value

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

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 (University of Notre Dame; <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.

Kelley, K. & Maxwell, S. E. (2003). Sample size for Multiple Regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*, 305–321.

Kelley, K. & Maxwell, S. E. (2008). Power and accuracy for omnibus and targeted effects: Issues of sample size planning with applications to Multiple Regression. *Handbook of Social Research Methods*, J. Brannon, P. Alasuutari, and L. Bickman (Eds.). New York, NY: Sage Publications.

Smithson, M. (2003). *Confidence intervals*. New York, NY: 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*, 164–182.

See Also

`ss.aipe.reg.coef`, `conf.limits.nct`, `ci.reg.coef`, `ci.src` `ci.reg.coef` *Confidence interval for a regression coefficient*

Description

A function to calculate a confidence interval around the population regression coefficient of interest using the standard approach and the noncentral approach when the regression coefficients are standardized.

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,
Suppress.Statement=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 `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`, specifying whether or not the noncentral approach to confidence intervals should be used
`Suppress.Statement`	`TRUE`/`FALSE` statement specifying whether or not a statement should be printed that identifies the type of confidence interval formed
`...`	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

Returns the confidence limits specified 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.

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 MBESS 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 (University of Notre Dame; `<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*, 305–321.

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

Smithson, M. (2003). *Confidence intervals*. New York, NY: Sage Publications.

See Also

`ss.aipe.reg.coef`, `conf.limits.nct`, `ci.rc`, `ci.src`

`ci.reliability` *Confidence Interval for a Reliability Coefficient*

Description

A function to calculate the point estimate and confidence interval for a reliability coefficient (alpha, omega, and variations thereof). Please see the many options; the defaults may not be best for your situtaion. See Kelley and Pornprasertmanit (in press) for recomendation and a discussion of the methods, where they ultimately recommend the bias corrected and accelerated bootstrap(`"interval.type=bca"` with categorical omega (`type=hierarchical`).

Usage

```
ci.reliability(data = NULL, S = NULL, N = NULL, aux = NULL,
type = "omega", interval.type = "mlr", B = 10000, conf.level = 0.95)
```

Arguments

`data`	The dataset that the reliability coefficient is obtained from. Real data set is required for categorical omega. Also, real data set is required for bootstrap confidence intervals or asymptotic distribution free confidence interval
`S`	Symmetric covariance matrix. Correlation matrix can be specified here but not recommended because, in the function, Confirmatory Factor Analysis (CFA) is analyzed based on covariance matrix.
`N`	The total sample size. Sample size is needed only that `S` is specified.
`aux`	The names of auxiliary variables. Auxiliary variables will not be used as a composite but they will be used to handle missing observations. Note that full information maximum likelihood is used if auxiliary variables are specified. See auxiliary for further details.
`type`	The type of reliability coefficient to be calculated: `"alpha"` or `1` for coefficient alpha analyzed by the formula proposed by Cronbach (1951), `"alpha-cfa"` or `2` for coefficient alpha analyzed by CFA with tau-equivalence (method of estimator depending on confidence interval method but none of them is unweighted least square so technically the result is not equal to the formula from Cronbach), `"omega"` for coefficient omega, `"hierarchical"` for hierarchical omega, `"categorical"` for categorical omega. The default is to use hierarchical omega for continuous items and categorical omega for categorical items.
`interval.type`	There are 13 options for the methods. See `details` below. Based on our simulation studies (Kelley and Pornprasertmanit, in press), bias corrected and accelerated bootstrap, `"bca"`, is recommended for categorical omega. Any bootstrap approaches (e.g., `"bca"` or `"perc"`) are recommended for hierarchical omega, coefficient omega, and coefficient alpha.
`B`	the number of bootstrap replications
`conf.level`	the confidence level (i.e., 1-Type I error rate)

Details

When coefficient alpha is used, the measurement model is assumed to be true-score equivalent (or tau equivalent) model such that factor loadings are equal across items. When the coefficient omega, hierarchical omega, and categorical omega are used, the measurement model is assumed to be congeneric model (i.e., one-factor confirmatory factor analysis model). Coefficient omega assumes that a model fits data perfectly so the variance of the composite scores is calculated from model-implied covariance matrix. However, hierarchical omega allows a model to not fit data perfectly (Kelley and Pornprasertmanit, in press). Categorical omega is a method to calculate coefficient omega for categorical items (Green and Yang, 2009). That is, categorical omega is estimated by the parameter estimates from CFA for categorical items. If coefficient omega or hierarchical omega is used, CFA for continuous items is used, which is not appropriate for categorical items.

If researchers wish to make the measurement model with all parallel items (equal factor loadings and equal error variances), users can specify it by setting `interval.type = "parallel"` and `type = "alpha"` or `type = "alpha-cfa"`. See McDonald (1999) for the assumptions of each of these models.

The list below shows all methods to find the confidence interval of reliability.

1. `"none"` or `0` to not find any confidence interval
2. `"parallel"` or `11` to assume that the items are parallel and analyze confidence interval based on wald confidence interval (see van Zyl, Neudecker, & Nel, 2000, Equation 22; also referred as the asymptotic method of Koning & Franses, 2003).
3. `"feldt"` or `12` is based on that $\frac{1-\alpha}{1-\hat{\alpha}}$ is distributed as F distribution with the degree of freedoms of $N-1$ and $(N-1)\times(p-1)$ (Feldt, 1965).
4. `"siotani"` or `13` is the same as the `"feldt"` method but using the degree of freedoms of N and $N\times(p-1)$ (Siotani, Hayakawa, & Fujikoshi, 1985; van Zyl et al., 2000, Equations 7 and 8; also referred as the exact method of Koning & Franses, 2003).
5. `"fisher"` or `21` for the Fisher's z transformation on the correlation coefficient approach, $z = 0.5 \times \log\frac{1+\alpha}{1-\alpha}$, directly on the coefficient alpha and find confidence interval of transformed scale (Fisher, 1950). The variance of the z is $\frac{1}{N-3}$ where N is the total sample size.
6. `"bonett"` or `22` for the Fisher's z transformation on the intraclass correlation approach with the variance of $\frac{2p}{(N-2)(p-1)}$ (Bonett, 2002, Equation 6).
7. `"hakstian"` or `23` uses the cube root transformation and assumes normal distribution on the cube root transformation (Hakstian & Whalen, 1976). The variance of the transformed reliability is based on the degrees of freedom in the `"feldt"` method.
8. `"hakstianbarchard"` or `24` uses a correction of the violation of compound symmetry of covariance matrix by adjusting the degrees of freedom in the `"hakstian"`. This correction is used for the inference in type 12 sampling (both persons and items are sampled from the population of persons and items) See Hakstian and Barchard (2000) for further details.
9. `"icc"` or `25` for the Fisher's z transformation on the intraclass correlation approach, $z = \log 1-\alpha$. The variance of the z is $\frac{2p}{N(p-1)}$ where p is the number of items (Fisher, 1991, p. 221; van Zyl et al., 2000, p. 277).
10. `"ml"` or `31` or `normal-theory` to analyze the confidence interval based on normal-theory approach (or multivariate delta method). See van Zyl, Neudecker, & Nel (2000, Equation 21) for the confidence interval of coefficient alpha (also be referred as Iacobucci & Duhachek's, 2003, method). See Raykov (2002) for details for coefficient omega. If users use `type="alpha-cfa"`, the `sem` package will be used to obtain parameter estimates and standard errors used for the formula proposed by Raykov (2002).
11. `"mll"` or `32` to analyze the confidence interval based on normal-theory approach as above. However, the point estimate and standard error were used to build confidence interval using logistic transformation as the note below.
12. `"mlr"` or `33` to analyze the confidence interval based on normal-theory approach (or multivariate delta method). However, the estimation method uses robust standard errors (Satorra and Bentler, 2000). This is the default estimation apporach (but see Kelley and Pornprasertmanit (in press) who recomend the BCa bootstrap [which is `bca`])
13. `"mlrl"` or `34` to analyze he confidence interval based on normal-theory approach using robust standard error and logistic transformation (see below).

14. "adf" or 35 for asymtotic distribution-free method (see Maydeu-Olivares, Coffman, & Hartman, 2007 for further details for coefficient omega; we use phantom variable approach, Cheung, 2009, and "WLS" estimator for coefficient omega, Browne, 1984, in the lavaan package, Rosseel, 2012).
15. "adfl" or 36 to use asymptotic distribution-free method to derive standard error and parameter estimate. Then, logistic transformation is used to build confidence interval (see below).
16. "ll" or 37 for profile likelihood-based confidence interval of both reliability coefficients (Cheung, 2009) analyzed by the OpenMx package (Boker et al., 2011)
17. "bsi" or 41 for standard bootstrap confidence interval which finds the standard deviation across the bootstrap estimates, multiply the standard deviation by critical value, and add and subtract from the reliability estimate.
18. "bsil" or 42 to use standard bootstrap confidence interval. However, logistic transformation is used to build confidence interval.
19. "perc" or 43 for percentile bootstrap confidence interval.
20. "bca" or 44 for bias-corrected and accelerated bootstrap confidence interval.

The logistic transformation (Browne, 1982) is applicable for "ml", "mlr", "adf", and "bsi" as "mll", "mlrl", "adfl", and "bsil". The logistic transformation does not assume that the sampling distribution of reliability is symmetric. It acknowledges the fact that reliability ranges from 0 and 1. Logistic transformation is applied to the reliability estimates. Confidence interval is established for the transformed value. The lower and upper bounds of the transformed value is translated back to the reliability estimates. See Browne (1982) or Kelley and Pornprasertmanit (in press) for further details.

Note that not all confidence interval methods are available for all types of reliability and all types of input. For example, bootstrap confidence intervals are not available for covariance matrix input. Parallel confidence intervals are not available for hierarchical omega. We provided appropriate error messages for all impossible combinations.

Value

est	The estimated reliability coefficient
se	The standard error of the reliability coefficient. If the bootstrap methods are used, this value represents the standard deviation across bootstrap estimates.
ci.lower	The lower bound of the computed confidence interval
ci.upper	The upper bound of the computed confidence interval
conf.Level	The confidence level (i.e., 1 - Type I error rate)
type	The type of estimated reliability coefficient (alpha or omega)
interval.type	The method used to find confidence interval

Note

This function is not compatible with code from MBESS Version 3.

Author(s)

Sunthud Pornprasertmanit (Texas Tech University; <sunthud.pornprasertmanit@ttu.edu>) and Ken Kelley (University of Notre Dame; <kkelley@nd.edu>. The previous version was written by Keke Lai (University of California-Merced), Leann J. Terry (while at Indiana University), and Ken Kelley

References

Boker, S., M., N., Maes, H., Wilde, M., Spiegel, M., Brick, T., et al. (2011). OpenMx: An open source extended structural equation modeling framework. *Psychometrika, 76,* 306–317.

Bonett, D. G. (2002). Sample size requirements for testing and estimating coefficient alpha. *Journal of Educational and Behavioral Statistics, 27,* 335–340.

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). Asymptotic distribution free methods in the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology, 24,* 445–455.

Cheung, M. W.-L. (2009). Constructing approximate confidence intervals for parameters with structural constructing approximate confidence intervals for parameters with structural equation models. *Structural Equation Modeling, 16,* 267–294.

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.

Fisher, R. A. (1991). Statistical methods for research workers. In J.H. Bennett (Ed.), *Statistical methods, experimental design, and scientific inference.* Oxford: Oxford University Press.

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

Hakstian, A. R., & Whalen, T. E. (1976). A k-sample significance test for independent alpha coefficients. *Psychometrika, 41,* 219–231.

Iacobucci, D., & Duhachek, A. (2003). Advancing alpha: measuring reliability with confidence. *Journal of Consumer Psychology, 13,* 478–487.

Kelley, K. & Pornprasertmanit, P. (in press). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for homogeneous composite measures. *Psychological Methods.*

Koning, A. J., & Franses, P. H. (2003). *Confidence intervals for Cronbach's coefficient alpha values* (ERIM Report Series Ref. No. ERS-2003-041-MKT). Rotterdam, The Netherlands: Erasmus Research Institute of Management.

Maydeu-Olivares, A., Coffman, D. L., & Hartmann, W. M. (2007). Asymptotically distribution-free (ADF) interval estimation of coefficient alpha. *Psychological Methods, 12,* 157–176.

McDonald, R. P. (1999). *Test theory: A unified approach.* Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Raykov, T. (2002). Analytic estimation of standard error and confidence interval for scale reliability. *Multivariate Behavioral Research, 37,* 89–103.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48,* 1–36.

Satorra, A. & Bentler, P. M. (2001). A scaled difference chi-square test statistic for moment structure analysis. *Psychometrika, 66,* 507–514.

Siotani, M., Hayakawa, T., & Fujikoshi, Y. (1985). *Modem multivariate statistical analysis: A graduate course and handbook*. Columbus, Ohio: American Sciences Press.

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.

Yuan, K. & Bentler, P. M. (2002). On robustness of the normal-theory based asymptotic distributions of three reliability coefficient estimates. *Psychometrika, 67* (2), 251–259.

See Also

`CFA.1`; `sem`; `lavaan`

Examples

```
# Use this function for the attitude dataset (ignoring the overall rating variable)
# ci.reliability(data=attitude[,-1], type = "omega", interval.type = "mlrl")

# ci.reliability(data=attitude[,-1], type = "alpha", interval.type = "ll")


## Forming a hypothetical population covariance matrix
# Pop.Cov.Mat <- matrix(.3, 9, 9)
# diag(Pop.Cov.Mat) <- 1
# ci.reliability(S=Pop.Cov.Mat, N=50, type="alpha", interval.type = "bonett")
```

---

ci.rmsea — *Confidence interval for the population root mean square error of approximation*

Description

Confidence interval for the population root mean square error of approximation (RMSEA).

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

Provides a confidence interval for the population root mean square error of approximation (RMSEA) using the noncentral chi-square distribution (e.g., Steiger & Lind, 1980).

Value

returns the upper and lower limit as well as the observed value of the RMSEA.

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`)

References

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.

`ci.sc` *Confidence Interval for a Standardized Contrast in a Fixed Effects ANOVA*

Description

Function to obtain the confidence interval for a standardized contrast in a fixed effects analysis of variance context.

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

`Lower.Conf.Limit.Standardized.Contrast`
: the lower confidence limit for the standardized contrast

`Standardized.contrast`
: standardized contrast

`Upper.Conf.Limit.Standardized.Contrast`
: the upper confidence limit for the standardized contrast

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 the error variance and not its square root (i.e., use the variance of the standard deviation of the errors). Be sure to use the standard deviations of errors for `s.anova` and `s.ancova`, not the square of these values (i.e., do not use the variance of the errors).

Be sure to use fractional *c*-weights when doing complex contrasts (not integers) to specify `c.weights`. For exmaple, 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 (University of Notre Dame; `<KKelley@ND.Edu>`)

References

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

Lai, K., & Kelley, K. (2007). Sample size planning for standardized ANCOVA and ANOVA contrasts: Obtaining narrow confidence intervals. *Manuscript submitted for publication*.

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*, 164–182.

See Also

`conf.limits.nct`, `ci.src`, `ci.smd`, `ci.smd.c`, `ci.sm`, `ci.c`

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,
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)

Value

`standardizer`	the divisor used in the standardization
`psi.limit.lower`	the lower confidence limit of the standardized contrast
`psi`	the estimated contrast
`psi.limit.upper`	the upper confidence limit of the standardized contrast

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)

Keke Lai (University of California–Merced) and Ken Kelley `<kkelley@nd.edu>`

References

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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.

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.

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`

Examples

```
# Maxwell & Delaney (2004, pp. 428--468) 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)
```

`ci.sm` *Confidence Interval for the Standardized Mean*

Description

Function to obtain the exact confidence interval for the standardized mean.

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 `conf.limits.nct` function in MBESS.

Value

`Lower.Conf.Limit.Standardized.Mean`
: lower confidence limit of the standardized mean

`Standardized.Mean`
: standardized mean

`Upper.Conf.Limit.Standardized.Mean`
: upper confidence limit of the standardized mean

Note

The standardized mean is the mean divided by the standard deviation.

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`)

References

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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

conf.limits.nct

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 limits for the standardized mean difference.*

Description

Function to calculate the confidence limits for the population standardized mean difference using the square root of the pooled variance as the divisor. This function is thus used to determine the confidence bounds for the population quantity of what is generally referred to as Cohen's *d* (delta being that population quantity).

Usage

```
ci.smd(ncp=NULL, smd=NULL, n.1=NULL, n.2=NULL, conf.level=.95,
alpha.lower=NULL, alpha.upper=NULL, tol=1e-9, ...)
```

Arguments

ncp	is the estimated noncentrality parameter, this is generally the observed *t*-statistic from comparing the two groups and assumes homogeneity of variance
smd	is the standardized mean difference (using the pooled standard deviation in the denominator)
n.1	is the sample size for Group 1
n.2	is the sample size for Group 2
conf.level	is the confidence level (1-Type I error rate)
alpha.lower	is the Type I error rate for the lower tail

`alpha.upper`	is the Type I error rate for the upper tail
`tol`	is the tolerance of the iterative method for determining the critical values
`...`	allows one to potentially include parameter values for inner functions

Value

`Lower.Conf.Limit.smd`
The lower bound of the computed confidence interval

`smd` The standardized mean difference

`Upper.Conf.Limit.smd`
The upper bound of the computed confidence interval

Warning

This function uses `conf.limits.nct`, 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 `conf.limits.nct` 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 (University of Notre Dame; <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*, 532–574.

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 2*, 107–128.

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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*, 258–287.

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`, `conf.limits.nct`

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

Function to calculate the confidence limits for the standardized mean difference using the control group standard deviation as the divisor (Glass's *g*).

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`	is the estimated noncentrality parameter, this is generally the observed *t*-statistic from comparing the control and experimental group (assuming homogeneity of variance)
`smd.c`	is the standardized mean difference (using the control group standard deviation in the denominator)
`n.C`	is the sample size for the control group
`n.E`	is the sample size for experimental group
`conf.level`	is the confidence level (1-Type I error rate)
`alpha.lower`	is the Type I error rate for the lower tail
`alpha.upper`	is the Type I error rate for the upper tail
`tol`	is the tolerance of the iterative method for determining the critical values
`...`	Potentially include parameter for inner functions

Value

`Lower.Conf.Limit.smd.c`	The lower bound of the computed confidence interval
`smd.c`	The standardized mean difference based on the control group standard deviation
`Upper.Conf.Limit.smd.c`	The upper bound of the computed confidence interval

Warning

This function uses `conf.limits.nct`, 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 `conf.limits.nct` 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 (University of Notre Dame; <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*, 532–574.

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, 2*, 107–128.

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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`, `conf.limits.nct`

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

Function to obtain the exact confidence interval for the signal-to-noise ratio (i.e., the variance of the specific factor over the error variance).

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 noncentality 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 `conf.limits.ncf` function in MBESS, which is used internally within this function.

Value

Returns the confidence limits for the signal-to-noise ratio.

`Lower.Limit.Signal.to.Noise.Ratio`
lower limit for signal to noise ratio

`Upper.Limit.Signal.to.Noise.Ratio`
upper limit for 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 (University of Notre Dame; <KKelley@ND.Edu>)

References

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

Fleishman, A. I. (1980). Confidence intervals for correlation ratios. *Educational and Psychological Measurement, 40*, 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*, 164–182.

See Also

`ci.srsnr`, `conf.limits.ncf`

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 11.2213. 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

Function to obtain the confidence interval for a standardized regression coefficient.

Usage

```
ci.src(beta.k = NULL, SE.beta.k = NULL, N = NULL, K = NULL, R2.Y_X = NULL,
R2.k_X.without.k = NULL, conf.level = 0.95, R2.Y_X.without.k = NULL,
t.value = NULL, b.k = NULL, SE.b.k = NULL, s.Y = NULL, s.X = NULL,
alpha.lower = NULL, alpha.upper = NULL, Suppress.Statement = FALSE, ...)
```

Arguments

`beta.k`	the standardized regression coefficient
`SE.beta.k`	the standard error of the standarized regression coefficient
`N`	sample size

`K`	the number of predictors
`R2.Y_X`	the squared multiple correlation coefficient predicting Y from the k predictor variables
`R2.k_X.without.k`	the squared multiple correlation coefficient predicting the kth 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.k`	the squared multiple correlation coefficient predicting Y from the p-1 predictor variable with the kth predictor of interest excluded
`t.value`	the t-value evaluating the null hypothesis that the population regression coefficient for the kth predictor equals zero
`b.k`	the unstandardized regression coefficient
`SE.b.k`	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
`Suppress.Statement`	`TRUE` or `FALSE` statement specifying whether or not a statement should be printed that identifies the type of confidence interval formed
`...`	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.k`.

Value

Returns the confidence limits specified 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.

Note

This function calls upon `ci.reg.coef` in MBESS, 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 (University of Notre Dame; <KKelley@ND.Edu>)

References

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

Kelley, K., & Maxwel, S. E. (2003). Sample size for Multiple Regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*, 305–321.

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

Smithson, M. (2003). *Confidence intervals*. New York, NY: 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*, 164–182.

See Also

`ss.aipe.reg.coef`, `conf.limits.nct`, `ci.reg.coef`, `ci.rc`

---

ci.srsnr — *Confidence Interval for the Square Root of the Signal-To-Noise Ratio*

Description

Function to calculate the exact confidence interval for the square root of the signal-to-noise ratio.

Usage

```
ci.srsnr(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`).

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 noncentality parameter to the confidence limits for square root of signal-to-noise ratio. The confidence interval for noncentral *F* parameter can be abtained from function `conf.limits.ncf` in MBESS.

Value

Returns the square root of the confidence limits for the signal to noise ratio.

`Lower.Limit.of.the.Square.Root.of.the.Signal.to.Noise.Ratio`
: lower limit of the square root of the signal to noise ratio

`Upper.Limit.of.the.Square.Root.of.the.Signal.to.Noise.Ratio`
: upper limit of the square root of the signal to noise ratio

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`)

References

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

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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*, 164–182.

See Also

`ci.snr`, `conf.limits.ncf`

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 11.2213.
## This exmaple 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)
```

`conf.limits.nc.chisq` *Confidence limits for noncentral chi square parameters*

Description

Function to determine the noncentral parameter that leads to the observed `Chi.Square`-value, so that a confidence interval for the population noncentral chi-squrae value can be formed.

Usage

```
conf.limits.nc.chisq(Chi.Square=NULL, conf.level=.95, df=NULL,
alpha.lower=NULL, alpha.upper=NULL, tol=1e-9, Jumping.Prop=.10)
```

Arguments

`Chi.Square`	the observed chi-square value
`conf.level`	the desired degree of confidence for the interval
`df`	the degrees of freedom
`alpha.lower`	Type I error for the lower confidence limit
`alpha.upper`	Type I error for the upper confidence limit
`tol`	tolerance for iterative convergence
`Jumping.Prop`	Value used in the iterative scheme to determine the noncentral parameters necessary for confidence interval construction using noncentral chi square-distributions (`0 < Jumping.Prop < 1`)

Details

If the function fails (or if a function relying upon this function fails), adjust the `Jumping.Prop` (to a smaller value).

Value

`Lower.Limit`	Value of the distribution with `Lower.Limit` noncentral value that has at its specified quantile `Chi.Square`
`Prob.Less.Lower`	Proportion of cases falling below `Lower.Limit`
`Upper.Limit`	Value of the distribution with `Upper.Limit` noncentral value that has at its specified quantile `Chi.Square`
`Prob.Greater.Upper`	Proportion of cases falling above `Upper.Limit`

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>); Keke Lai (University of California–Merced)

See Also

conf.limits.nct, conf.limits.ncf

Examples

```
# A typical call to the function.
conf.limits.nc.chisq(Chi.Square=30, conf.level=.95, df=15)

# A one sided (upper) confidence interval.
 conf.limits.nc.chisq(Chi.Square=30, alpha.lower=0, alpha.upper=.05,
 conf.level=NULL, df=15)
```

---

conf.limits.ncf *Confidence limits for noncentral F parameters*

Description

Function to determine the noncentral parameter that leads to the observed F-value, so that a confidence interval around the population F-value can be conducted. Used for forming confidence intervals around noncentral parameters (given the monotonic relationship between the F-value and the noncentral value).

Usage

```
conf.limits.ncf(F.value = NULL, conf.level = .95, df.1 = NULL,
df.2 = NULL, alpha.lower = NULL, alpha.upper = NULL, tol = 1e-09,
Jumping.Prop = 0.1)
```

Arguments

F.value	the observed F-value
conf.level	the desired degree of confidence for the interval
df.1	the numerator degrees of freedom
df.2	the denominator degrees of freedom
alpha.lower	Type I error for the lower confidence limit
alpha.upper	Type I error for the upper confidence limit
tol	tolerance for iterative convergence
Jumping.Prop	Value used in the iterative scheme to determine the noncentral parameters necessary for confidence interval construction using noncentral F-distributions (0 < Jumping.Prop < 1) (users should not need to change this value)

Details

This function is the relied upon by the `ci.R2` and `ss.aipe.R2`. If the function fails (or if a function relying upon this function fails), adjust the `Jumping.Prop` (to a smaller value).

Value

- `Lower.Limit` — Value of the distribution with `Lower.Limit` noncentral value that has at its specified quantile `F.value`
- `Prob.Less.Lower` — Proportion of cases falling below `Lower.Limit`
- `Upper.Limit` — Value of the distribution with `Upper.Limit` noncentral value that has at its specified quantile `F.value`
- `Prob.Greater.Upper` — Proportion of cases falling above `Upper.Limit`

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>); Keke Lai (University of Califonia-Merced)

See Also

`ss.aipe.R2`, `ci.R2`, `conf.limits.nct`

Examples

```
conf.limits.ncf(F.value = 5, conf.level = .95, df.1 = 5,
df.2 = 100)

# A one sided confidence interval.
conf.limits.ncf(F.value = 5, conf.level = NULL, df.1 = 5,
df.2 = 100, alpha.lower = .05, alpha.upper = 0, tol = 1e-09,
Jumping.Prop = 0.1)
```

`conf.limits.nct` *Confidence limits for a noncentrality parameter from a t-distribution*

Description

Function to determine the noncentrality parameters necessary to form a confidence interval around the population noncentrality parameter and related parameters.

Usage

```
conf.limits.nct(ncp, df, conf.level = 0.95, alpha.lower = NULL,
alpha.upper = NULL, t.value, tol = 1e-09, sup.int.warns = 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`	is the tolerance of the iterative method for determining the critical values.
`sup.int.warns`	Suppress internal warnings (from internal functions): `TRUE` or `FALSE`
`...`	allows one to potentially include parameter values for inner functions

Details

Function for finding the upper and lower confidence limits for a noncentral parameter from a noncentral *t*-distribution with `df` degrees of freedom. This function is especially helpful when forming confidence intervals around standardized mean differences (i.e., Cohen's *d*; Glass's *g*; Hedges' *g*), standardized regression coefficients, and coefficients of variations. The `Lower.Limit` and the `Upper.Limit` values correspond to the noncentral parameters of a *t*-distribution with `df` degrees of freedom whose upper and lower tails contain the desired proportion of the respective noncentral *t*-distribution. When `ncp` is zero, the `Lower.Limit` and `Upper.Limit` are simply the desired quantiles of the central *t*-distribution with `df` degrees of freedom.

Note that the confidence interval limit(s) are found twice, using two different methods. The first method uses the `optimize` function, whereas the second method uses the `nlm` function. The best of the two methods, if not equal and numerically exact, is taken. This does not concern the user.

Value

`Lower.Limit`	Value of the distribution with `Lower.Limit` noncentral value that has at its specified quantile `F.value`
`Prob.Less.Lower`	Proportion of the distribution beyond (i.e., less than) `Lower.Limit`
`Upper.Limit`	Value of the distribution with `Upper.Limit` noncentral value that has at its specified quantile `F.value`
`Prob.Greater.Upper`	Proportion of the distribution beyond (i.e., larger than) `Upper.Limit`

Warning

At the present time, the largest `ncp` that R can accurately handle is 37.62.

Author(s)

Ken Kelley (University of Notre Dame; `<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*, 532–574.

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.

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

Steiger, J. & Fouladi, T. (1997). Noncentrality interval estimation and the evaluation of statistical models. In L. Harlow, S. Muliak, & J. Steiger (Eds.), *What if there were no significance tests?*. Mahwah, NJ: Lawrence Erlbaum.

See Also

`pt`, `qt`, `ci.smd`, `ci.smd.c`, `ss.aipe`, `conf.limits.ncf`, `conf.limits.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:
conf.limits.nct(ncp=2.83, df=126, conf.level=.95)

# Modifying the above example so that a nonsymmetric 95% confidence interval
# can be formed:
conf.limits.nct(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:
conf.limits.nct(ncp=2.83, df=126, alpha.lower=0, alpha.upper=.05,
conf.level=NULL)
```

`Cor.Mat.Lomax` *Correlation matrix for Lomax (1983) data set*

Description

Correlation matrix for Lomax (1983) data set

Usage

```
data(Cor.Mat.Lomax)
```

Details

Variables 1 through 14 in the correlation matrix are, respectively:

Variables
(1) DRS-consonant sounds
(2) DRS-consonant blends and diagraphs
(3) DRS-common syllables or phonograms
(4) DRS-blending
(5) WRAT-total raw score
(6) DRS-total correct both lists
(7) DRS-total words read correct oral
(8) DRS-wpm first oral passage
(9) DRS-wpm first silent passage
(10) DRS-mean wpm oral passages read
(11) DRS-mean wpm silent passages read
(12) DRS-total correct oral comprehension
(13) DRS-total correct silent comprehension
(14) CTBS-comprehension ESS scores

DRS refers to Diagnostic Reading Scales, WRAT refers to Wide Range Achievement Test, and CTBS refers to Comprehensive Tests of basic skills.

The model was designed to study the causal relationship between the phonological, word recognition, reading rate, and comprehension components of the reading process. There are four latent variables in the model: (a) phonological; (b) word recognition; (c) reading rate; (d) reading comprehension.

Phonological is indicated by (a) DRS-consonant sounds; (b) DRS-consonant blends and diagraphs; (c) DRS-common syllables or phonograms; (d) DRS-blending.

Word recognition is indicated by (a) WRAT-total raw score; (b) DRS-total correct both lists; (c) DRS-total words read correct oral

Reading rate is indicated by (a) DRS-wpm first oral passage; (b) DRS-wpm first silent passage; (c) DRS-mean wpm oral passages read; (d) DRS-mean wpm silent passages read.

Reading comprehension is indicated by (a) DRS-total correct oral comprehension; (b) DRS-total correct silent comprehension; (c) CTBS-comprehension ESS scores.

Source

Lomax, R. G. (1983). Applying structural modeling to some component processes of reading comprehension development. *Journal of Experimental Education, 52* (1), 33–40.

References

Lomax, R. G. (1983). Applying structural modeling to some component processes of reading comprehension development. *Journal of Experimental Education, 52* (1), 33–40.

`Cor.Mat.MM` *Correlation matrix for Maruyama & McGarvey (1980) data set*

Description

Correlation matrix for Maruyama & McGarvey (1980) data set

Usage

```
data(Cor.Mat.MM)
```

Details

Variables 1 through 13 in the correlation matrix are, respectively:

Variables
(1) seating popularity
(2) playground popularity
(3) schoolwork popularity
(4) verbal achievement
(5) verbal grades
(6) Duncan SEI
(7) education of head of house
(8) No. of rooms over No. of persons
(9) Raven Progressive Matrices
(10) Peabody PVT
(11) father's evaluation
(12) mothers evaluation
(13) teacher's evaluation

The model was designed to examine whether acceptance by significant others (i.e., parents, teachers, and peers) causes improved scholastic achievement. There are five latent variables in the model: (a) SES, socio-economic status; (b) ABL, academic ability; (c) ACH, achievement; (d) ASA, acceptance by significant adults; (e) APR, acceptance by peers.

SES is indicated by (a) SEI, Duncan Socioeconomic Index of Occupations; (b) EDHH, educational attainment of the head of the household; (c) R/P, ratio of rooms in the house to persons living in the house.

ACH is indicated by (a) VACH, standardized verbal test scores; (b) VGR, verbal grades.

ABL is indicated by (a) PEA, Peabody Picture Vocabulary Test; (b) RAV, Raven Progressive Matrices.

ASA is indicated by (a) FEV, father's evaluation; (b) MEV, mother's evaluation; (c) TEV, teacher's evaluation.

APR is indicated by (a) PPOP, playground popularity; (b) SPOP, seating popularity; (c) WPOP, schoolwork popularity.

Source

Maruyama, G., & McGarvey, B. (1980). Evaluating causal models: An application of maximum-likelihood analysis of structural equations. *Psychological Bulletin, 87* (3), 502–512.

References

Maruyama, G., & McGarvey, B. (1980). Evaluating causal models: An application of maximum-likelihood analysis of structural equations. *Psychological Bulletin, 87* (3), 502–512.

---

cor2cov — *Correlation Matrix to Covariance Matrix Conversion*

Description

Function to convert a correlation matrix to a covariance matrix.

Usage

```
cor2cov(cor.mat, sd, discrepancy=1e-5)
```

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 neighborhood of 1, such that numbers on the main diagonal of the correlation matrix will be considered as equal to 1 if they fall in this neighborhood

Details

The correlation matrix to convert can be either symmetric or triangular. The covariance matrix returned is always a symmetric matrix.

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, eg, 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 (University of Notre Dame; <KKelley@ND.Edu>), Keke Lai covmat.from.cfm *Covariance matrix from confirmatory (single) factor model.*

Description

Function calculates a covariance matrix using the specified Lambda and Psi.Square values from a confirmatory factor model approach (McDonald, 1999).

Usage

```
covmat.from.cfm(Lambda, Psi.Square, tol.det = 1e-05)
```

Arguments

Lambda	the vector of population factor loadings
Psi.Square	the vector of population error variances
tol.det	the specified tolerance for the determinant

Value

Population.Covariance
: the population covariance matrix

True.Covariance
: the true covariance matrix

True.Covariance
: the error covariance matrix

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>); Leann Terry (Indiana University; <ljterry@Indiana.Edu>)

References

McDonald, R. P. (1999). *Test theory: A unified approach*. Mahwah, NJ: Erlbaum.

See Also

CFA.1;sem

Examples

```
# General Congeneric
# covmat.from.cfm(Lambda=c(.8, .9, .6, .8), Psi.Square=c(.6, .2, .1, .3), tol.det=.00001)


# True-score equivalent
# covmat.from.cfm(Lambda=c(.8, .8, .8, .8), Psi.Square=c(.6, .2, .1, .3), tol.det=.00001)
```

```
# Parallel
# covmat.from.cfm(Lambda=c(.8, .8, .8, .8), Psi.Square=c(.2, .2, .2, .2), tol.det=.00001)
```

---

cv *Function to calculate the regular (which is also biased) estimate of the coefficient of variation or the unbiased estimate of the coefficient of variation.*

Description

Returns the estimated coefficient of variation or the unbiased estimate of the coefficient of variation.

Usage

```
cv(C.of.V=NULL, mean=NULL, sd=NULL, N=NULL, unbiased=FALSE)
```

Arguments

`C.of.V`	Usual estimate of the coefficient of variation (`C.of.V=sd/mean`)
`mean`	observed mean
`sd`	observed standard deviation (based on `N-1` in the denominator of the variance)
`N`	sample size
`unbiased`	return the unbiased estimate of the coefficient of variation

Details

A function to calculate the usual estimate of the coefficient of variation or its unbiased estimate.

Value

Returns the estimated coefficient of variation (regular but biased estimate or unbiased estimate.

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`)

See Also

`ci.cv`

Examples

```
cv(mean=100, sd=15)
cv(mean=100, sd=15, N=50, unbiased=TRUE)
cv(C.of.V=.15, N=2, unbiased=TRUE)
```

`Expected.R2` *Expected value of the squared multiple correlation coefficient*

Description

Returns the expected value of the squared multiple correlation coefficient given the population squared multiple correlation coefficient, sample size, and the number of predictors

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

Returns the expected value of the squared multiple correlation coefficient.

Note

Uses package `gsl` and its `hyperg_2F1` function.

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`)

References

Olkin, I. & Pratt, J. W. (1958). Unbiased estimation of certain correlation coefficients. *Annals of Mathematical statistics, 29*, 201–211.

Stuart, A., Ord, J. K., & Arnold, S. (1999). *Kendall's advanced theory of statistics: Classical inference and the linear model* (Volume 2A, 2nd Edition). New York, NY: Oxford University Press.

See Also

`ss.aipe.R2`, `ci.R2`, `Variance.R2`

Examples

```
# library(gsl)
# 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)
```

F.and.R2.Noncentral.Conversion

Conversion functions from noncentral noncentral values to their corresponding and vice versa, for those related to the F-test and R Square.

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

```
Rsquare2F(R2 = NULL, df.1 = NULL, df.2 = NULL, p = NULL, N = NULL)

F2Rsquare(F.value = NULL, df.1 = NULL, df.2 = NULL)

Lambda2Rsquare(Lambda = NULL, N = NULL)

Rsquare2Lambda(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

Returns the converted value from the specified function.

Author(s)

Ken Kelley (University of Notre Dame, `<KKelley@ND.Edu>`)

See Also

`ss.aipe.R2`, `ci.R2`, `conf.limits.nct`, `conf.limits.ncf`

Examples

```
Rsquare2Lambda(R2=.5, N=100)
```

`Gardner.LD` *The Gardner learning data, which was used by L.R. Tucker*

Description

Repeated measures data on 24 participants, each with 21 trials (each trial based on 20 replications).

Usage

```
data(Gardner.LD)
```

Format

A data frame where the rows represent the timepoints for the individuals.

`ID` : a numeric vector

`Trial` : a numeric vector

`Score` : a numeric vector

`Group` : a numeric vector

Details

The 24 participants of this study were presented with 420 presentations of four letters where the task was to identify the next letter that was to be presented. Twelve of the participants (Group 1) were presented the letters S, L, N, and D with probabilities .70, .10, .10, and .10, respectively. The other 12 participants (Group 2) were presented the letter L with probability .70 and three other letters, each with a probability of .10. The 420 presentations were (arbitrarily it seems) grouped into 21 trials of 20 presentations. The score for each trial was the number of times the individual correctly guessed the dominant letter. The participants were naive to the probability that the letters would be presented. Other groups of individuals (although the data is not available) were tested under a different probability structure. The data given here is thus known as the 70-10-10-10 group from Gardner's paper. L. R. Tucker used this data set to illustrate methods for understanding change.

Source

Tucker, L. R. (1960). Determination of Generalized Learning Curves by Factor Analysis, Educational Testing Services, Princeton, NJ.

References

Gardner, R. A., (1958). Multiple-choice decision-behavior, *American Journal of Psychology*, 71, 710–717.

`HS.data` *Complete Data Set of Holzinger and Swineford's (1939) Study*

Description

The *complete* data set of scores of 301 subjects in 26 tests in Holzinger and Swineford's (1939) study.

Usage

```
data(HS.data)
```

Format

A data frame with 301 observations on the following 32 variables.

`id` subject's ID number

`Gender` subject's gender

`grade` the grade the subject is in

`agey` the year part of the subject's age

`agem` the month part of the subject's age

`school` the school the subject is from

`visual` scores on visual perception test, test 1

`cubes` scores on cubes test, test 2

`paper` scores on paper form board test, test 3

`flags` scores on lozenges test, test 4

`general` scores on general information test, test 5

`paragrap` scores on paragraph comprehension test, test 6

`sentence` scores on sentence completion test, test 7

`wordc` scores on word classification test, test 8

`wordm` scores on word meaning test, test 9

`addition` scores on add test, test 10

`code` scores on code test, test 11

`counting` scores on counting groups of dots test, test 12

`straight` scores on straight and curved capitals test, test 13

`wordr` scores on word recognition test, test 14

`numberr` scores on number recognition test, test 15

`figurer` scores on figure recognition test, test 16

`object` scores on object-number test, test 17

`numberf` scores on number-figure test, test 18

`figurew` scores on figure-word test, test 19

`deduct` scores on deduction test, test 20

`numeric` scores on numerical puzzles test, test 21

`problemr` scores on problem reasoning test, test 22

`series` scores on series completion test, test 23

`arithmet` scores on Woody-McCall mixed fundamentals, form I test, test 24

`paperrev` scores on additional paper form board test, test 25

`flagssub` scores on flags test, test 26

Details

Holzinger and Swineford (1939) data is widely cited, but generally only the Grant-White School data is used. The present dataset contains the complete data of Holzinger and Swineford (1939).

A total number of 301 pupils from Paster School and Grant-White School participated in Holzinger and Swineford's (1939) study. This study consists of 26 tests, which are used to measure the subjects' spatial, verbal, mental speed, memory, and mathematical ability.

The spatial tests consist of `visual`, `cubes`, `paper`, `flags`, `paperrev`, and `flagssub`. The test 25, paper form board test (`paperrev`), can be used as a subsitute for test 3, paper form board test (`paper`). The test 26, flags test (`flagssub`), is a possible substitute for test 4, lozenges test (`flags`).

The verbal tests consist of `general`, `paragrap`, `sentence`, `wordc`, and `wordm`.

The speed tests consist of `addition`, `code`, `counting`, and `straight`.

The memory tests consist of `wordr`, `numberr`, `figurer`, `object`, `numberf`, and `figurew`.

The mathematical-ability tests consist of `deduct`, `numeric`, `problemr`, `series`, and `arithmet`.

Source

Holzinger, K. J. and Swineford, F. A. (1939). A study in factor analysis: The stability of a bi-factor solution. *Supplementary Education Monographs, 48*. University of Chicago.

References

Holzinger, K. J. and Swineford, F. A. (1939). A study in factor analysis: The stability of a bi-factor solution. *Supplementary Education Monographs, 48*. University of Chicago.

`intr.plot` *Regression Surface Containing Interaction*

Description

To plot a three dimentional figure of a multiple regression surface containing one two-way interaction.

Usage

```
intr.plot(b.0, b.x, b.z, b.xz, x.min = NULL, x.max = NULL, z.min = NULL,
z.max = NULL, n.x = 50, n.z = 50, x = NULL, z = NULL, col = "lightblue",
hor.angle = -60, vert.angle = 15, xlab = "Value of X", zlab = "Value of Z",
ylab = "Dependent Variable", expand = 0.5, lines.plot=TRUE, col.line = "red",
line.wd = 2, gray.scale = FALSE, ticktype="detailed", ...)
```

Arguments

`b.0`	the intercept
`b.x`	regression coefficient for predictor x
`b.z`	regression coefficient for predictor z
`b.xz`	regression coefficient for the interaction of predictors x and z
`x.min, x.max, z.min, z.max`	ranges of x and z. The regression surface defined by these limits will be plotted.
`n.x`	number of elements in predictor vector x; number of points to be plotted on the regression surface; default is 50
`n.z`	number of elements in predictor vector z; number of points to be plotted on the regression surface; default is 50
`x`	a specific predictor vector `x`, used instead of `x.max` and `x.min`
`z`	a specific predictor vector `z`, used instead of `z.max` and `z.min`
`col`	color of the regression surface; default is lightbule
`hor.angle`	rotate the regression surface horizontally; default is -60 degree
`vert.angle`	rotate the regression surface vertically; default is 15 degree
`xlab`	title for the axis which the predictor `x` is on
`zlab`	title for the axis which the predictor `z` is on
`ylab`	title for the axis which the dependent `y` is on
`expand`	default is 0.5; expansion factor applied to the axis of the dependent variable. Often used with 0 < `expand` < 1 to shrink the plotting box in the direction of the dependent variable's axis.
`lines.plot`	whether or not to plot on the regression surface regression lines holding z at values 0, 1, -1, 2, -2 above the mean; default is `TRUE`.
`col.line`	the color of regression lines plotted on the regression surface; default is red

line.wd	the width of regression lines plotted on the regression surface; default is 2
gray.scale	whether or not to plot the figure black and white; default is FALSE
ticktype	whether the axes should be plotted with ("detailed") or without ("simple") tick marks
...	allows one to potentially include parameter values for inner functions

Details

The user can input either the limits of x and z, or specific x and z vectors, to draw the regression surface. If the user inputs simply the limits of the predictors, the function would generate predictor vectors for plotting. If the user inputs specific predictor vectors, the function would plot the regression surface based on those vectors.

Note

If the user enters specific vectors instead of the ranges of predictors, please make sure elements in those vectors are in ascending order. This is required by function persp, which is used within this function.

Author(s)

Keke Lai (University of California – Merced) and Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>)

References

Cohen, J., Cohen, P., West, S. G. and Aiken, L. S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (3rd ed.). Mahwah, NJ: Erlbaum.

See Also

intr.plot.2d, persp

Examples

```
##  A way to replicate the example given by Cohen et al. (2003) (pp. 258--263):
##  The regression equation with interaction is y=.2X+.6Z+.4XZ+2
##  To plot a regression surface and regression lines of Y on X holding Z
##  at -1, 0, and 1 standard deviation above the mean

x<- c(0,2,4,6,8,10)
z<-c(0,2,4,6,8,10)
intr.plot(b.0=2, b.x=.2, b.z=.6, b.xz=.4, x=x, z=z)

## input limits of the predictors instead of specific x and z predictor vectors
intr.plot(b.0=2, b.x=.2, b.z=.6, b.xz=.4, x.min=5, x.max=10, z.min=0, z.max=20)

intr.plot(b.0=2, b.x=.2, b.z=.6, b.xz=.4, x.min=0, x.max=10, z.min=0, z.max=10,
col="gray", hor.angle=-65, vert.angle=10)

##  To plot a black-and-white figure
```

```
intr.plot(b.0=2, b.x=.2, b.z=.6, b.xz=.4, x.min=0, x.max=10, z.min=0, z.max=10,
gray.scale=TRUE)

## to adjust the tick marks on the axes
intr.plot(b.0=2, b.x=.2, b.z=.6, b.xz=.4, x.min=0, x.max=10, z.min=0, z.max=10,
ticktype="detailed", nticks=8)
```

`intr.plot.2d` *Plotting Conditional Regression Lines with Interactions in Two Dimensions*

Description

To plot regression lines for one two-way interactions, holding one of the predictors (in this function, z) at values -2, -1, 0, 1, and 2 standard deviations above the mean.

Usage

```
intr.plot.2d(b.0, b.x, b.z, b.xz,x.min=NULL, x.max=NULL, x=NULL,
n.x=50, mean.z=NULL, sd.z=NULL, z=NULL,xlab="Value of X",
ylab="Dependent Variable", sd.plot=TRUE, sd2.plot=TRUE, sd_1.plot=TRUE,
sd_2.plot=TRUE, type.sd=2, type.sd2=3, type.sd_1=4, type.sd_2=5,
legend.pos="bottomright", legend.on=TRUE, ... )
```

Arguments

`b.0`	the intercept
`b.x`	regression coefficient for predictor x
`b.z`	regression coefficient for predictor z
`b.xz`	regression coefficient for the interaction of predictors x and z
`x.min,x.max`	the range of x used in the plot
`x`	a specific predictor vector x, used instead of `x.min` and `x.max`
`n.x`	number of elements in predictor vector x
`mean.z`	mean of predictor z
`sd.z`	standard deviation of predictor z
`z`	a specific predictor vector z, used instead of `z.min` and `z.max`
`xlab`	title for the axis which the predictor x is on
`ylab`	title for the axis which the dependent y is on
`sd.plot, sd2.plot, sd_1.plot, sd_2.plot`	whether or not to plot the regression line holding z at values 1, 2, -1, and -2 standard deviations above the mean, respectively. Default values are all `TRUE`.
`type.sd, type.sd2, type.sd_1, type.sd_2`	types of lines to be plotted holding z at values 1, 2, -1, and -2 standard deviations above the mean, respectively. Default are line type 2,3,4, and 5, respectively.

`legend.pos`	position of the legend; possible options are `"bottomright"`, `"bottom"`, `"bottomleft"`, `"left"`, `"center"`, `"right"`, `"topleft"`, `"top"`, and `"topright"`.
`legend.on`	whether or not to show the legend
`...`	allows one to potentially include parameter values for inner functions

Details

To input the predictor x, one can use either the limits of x (`x.max` and `x.min`) , or a specific vector x (`x`). To input the predictor z, one can use either the mean and standard deviation of z (`mean.z` and `sd.z`), or a specific vector z (`z`).

Note

Sometimes some of the regression lines are outside the default scope of the coordinates and thus cannot be seen; in such situations, one needs to, by entering additional arguments, adjust the scope to let proper sections of regression lines be seen. Refer to examples below for more details.

Author(s)

Keke Lai, Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>)

References

Cohen, J., Cohen, P., West, S. G. and Aiken, L. S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (3rd ed.). Mahwah, NJ: Erlbaum.

See Also

`intr.plot`

Examples

```
## A situation where one regression line is outside the default scope of the coordinates
intr.plot.2d(b.0=16, b.x=2.2, b.z=2.6, b.xz=.4, x.min=0, x.max=20, mean.z=0, sd.z=3)

## Adjust the scope of x and y axes so that proper sections of regression lines can be seen
intr.plot.2d(b.0=16, b.x=2.2, b.z=2.6, b.xz=.4, x.min=0, x.max=50, mean.z=0,
sd.z=3, xlim=c(0,50), ylim=c(-20,100) )

## Use specific vector(s) to define the predictor(s)
intr.plot.2d(b.0=16, b.x=2.2, b.z=2.6, b.xz=.4, x=c(1:10), z=c(0,2,4,6,8,10))

intr.plot.2d(b.0=16, b.x=2.2, b.z=2.6, b.xz=.4, x.min=0, x.max=20,
z=c(1,3,6,7,9,13,16,20), ylim=c(0,100))

## Change the position of the legend so that it does not block regression lines
intr.plot.2d(b.0=10, b.x=-.3, b.z=1, b.xz=.5, x.min=0, x.max=40, mean.z=-5, sd.z=3,
ylim=c(-100,100),legend.pos="topright" )
```

MBESS — *MBESS*

Description

MBESS Implements methods that useful in designing research and analyzing data, with particular emphasis on methods that are developed for or used within the behavioral, educational, and social sciences (broadly defined). That being said, many of the methods implemented within MBESS are applicable to a wide variety of disciplines. MBESS has a suite of functions for a variety of related topics, such as effect sizes, confidence intervals for effect sizes (including standardized effect sizes and noncentral effect sizes), sample size planning (from the accuracy in parameter estimation [AIPE], power analytic, equivalence, and minimum-risk point estimation perspectives), mediation analysis, various properties of distributions, and a variety of utility functions. MBESS (pronounced 'em-bes') was originally an acronym for "Methods for the Behavioral, Educational, and Social Sciences," but at this point MBESS contains methods applicable and used in a wide variety of fields and is an orphan acronym, in the sense that what was an acronym is now literally its name. MBESS has greatly benefited from others, see <http://nd.edu/~kkelley/site/MBESS.html> for a detailed list of those that have contributed and other details.

Details

Package:	MBESS
Type:	Package
Version:	4.3.0
Date:	2017-07-05
License:	GPL(>=2)

Please read the manual and visit the corresponding web site http://nd.edu/~kkelley/site/MBESS.html for information on the capabilities of the MBESS package. Feel free to contact me if there is a feature you would like to see added if it would complement the goals of the MBESS package.

Author(s)

Ken Kelley <<KKelley@ND.Edu>; http://www.nd.edu/~kkelley>

Maintainer: Ken Kelley <<KKelley@ND.Edu>; http://www.nd.edu/~kkelley>

mediation — *Effect sizes and confidence intervals in a mediation model*

Description

Automate the process of simple mediation analysis (one independent variable and one mediator) and effect size estimation for mediation models, as discussed in Preacher and Kelley (2011).

Usage

```
mediation(x, mediator, dv, S = NULL, N = NULL, x.location.S = NULL,
mediator.location.S = NULL, dv.location.S = NULL, mean.x = NULL,
mean.m = NULL, mean.dv = NULL, conf.level = 0.95,
bootstrap = FALSE, B = 10000, which.boot="both", save.bs.replicates=FALSE,
complete.set=FALSE)
```

Arguments

`x`	vector of the predictor/independent variable
`mediator`	vector of the mediator variable
`dv`	vector of the dependent/outcome variable
`S`	Covariance matrix
`N`	Sample size, necessary when a covariance matrix (`S`) is used
`x.location.S`	location of the predictor/independent variable in the covariance matrix (`S`)
`mediator.location.S`	location of the mediator variable in the covariance matrix (`S`)
`dv.location.S`	location of the dependent/outcome variable in the covariance matrix (`S`)
`mean.x`	mean of the `x` (independent/predictor) variable when a covariance matrix (`S`) is used
`mean.m`	mean of the `m` (mediator) variable when a covariance matrix (`S`) is used
`mean.dv`	mean of the `y/dv` (dependent/outcome) variable when a covariance matrix (`S`) is used
`conf.level`	desired level of confidence (e.g., .90, .95, .99, etc.)
`bootstrap`	`TRUE` or `FALSE`, based on whether or not a bootstrap procedure is performed to obtain confidence intervals for the various effect sizes
`B`	number of bootstrap replications when `bootstrap=TRUE` (e.g., 10000)
`which.boot`	which bootstrap method to use. It can be `Percentile` or `BCa`, or both
`save.bs.replicates`	Logical argument indicating whether to save the each bootstrap sample or not
`complete.set`	identifies if the function should report the estimated kappa.squarred (see below)

Details

Based on the work of Preacher and Kelley (2010) and works cited therein, this function implements (simple) mediation analysis in a way that automates much of the results that are generally of interest, where "simple" means one independent variable, one mediator, and one dependent variable. More specifically, three regression outputs are automated as is the calculation of effect sizes that are thought to be useful or potentially useful in the context of mediation. Much work on mediation models exists in the literature, which should be consulted for proper interpretation of the effect sizes, models, and meaning of results. The usefulness of effect size κ^2 was called into question by Wen and Fan (2015). Further, another paper by Lachowicz, Preacher, and Kelley (submitted) offers a better was of quantifying the effect size and it is developed for more complex models. Users are encouraged to use, instead of or in addition to this function, the upsilon function.

Value

`Y.on.X$Regression.Table`
: Regression table of `Y` conditional on `X`

`Y.on.X$Model.Fit`
: Summary of model fit for the regression of `Y` conditional on `X`

`M.on.X$Regression.Table`
: Regression table of `X` conditional on `M`

`M.on.X$Model.Fit`
: Summary of model fit for the regression of `X` conditional on `M`

`Y.on.X.and.M$Regression.Table`
: Regression table of `Y` conditional on `X` and `M`

`Y.on.X.and.M$Model.Fit`
: Summary of model fit for the regression of `Y` conditional on `X` and `M`

`Indirect.Effect`
: the product of $\hat{a} \times \hat{b}$, where $\hat{a}$ and $\hat{b}$ are the estimated coefficients of the path from the independent variable to the mediator and the path from the mediator to the dependent variable

`Indirect.Effect.Partially.Standardized`
: It is the indirect effect (see `Indirect.Effect` above) divided by the estimated standard deviation of `Y` (MacKinnon, 2008)

`Index.of.Mediation`
: Index of mediation (indirect effect multiplied by the ratio of the standard deviation of X to the standard deviation of Y) (Preacher and Hayes, 2008)

`R2_4.5`
: An index of explained variance see MacKinnon (2008, Eq. 4.5) for details

`R2_4.6`
: An index of explained variance see MacKinnon (2008, Eq. 4.6) for details

`R2_4.7`
: An index of explained variance see MacKinnon (2008, Eq. 4.7) for details

`Maximum.Possible.Mediation.Effect`
: the maximum attainable value of the mediation effect (i.e., the indirect effect), in the direction of the observed indirect effect, that could have been observed, conditional on the sample variances and on the magnitudes of relationships among some of the variables

`ab.to.Maximum.Possible.Mediation.Effect_kappa.squared`
: the proportion of the maximum possible indirect effect; Uses the indirect effect in the numerator with the maximum possible mediation effect in the denominator (Preacher & Kelley, 2010)

`Ratio.of.Indirect.to.Total.Effect`
: ratio of the indirect effect to the total effect (Freedman, 2001); also known as mediation ratio (Ditlevsen, Christensen, Lynch, Damsgaard, & Keiding, 2005); in epidemiological research and as the relative indirect effect (Huang, Sivaganesan, Succop, & Goodman, 2004); often loosely interpreted as the relative indirect effect

`Ratio.of.Indirect.to.Direct.Effect`
: ratio of the indirect effect to the direct effect (Sobel, 1982)

`Success.of.Surrogate.Endpoint`
: Success of a surrogate endpoint (Buyse & Molenberghs, 1998)

`SOS` shared over simple effects (SOS) index, which is the ratio of the variance in Y explained by both `X` and `M` divided by the variance in `Y` explained by `X` (Lindenberger & Potter, 1998)

`Residual.Based_Gamma`
A residual based index (Preacher & Kelley, 2010)

`Residual.Based.Standardized_gamma`
A residual based index that is standardized, where the scales of M and Y are removed by using standardized values of M and Y (Preacher & Kelley, 2010)

`ES.for.two.groups`
When X is 0 and 1 representing a two group structure, Hansen and McNeal's (1996) Effect Size Index for Two Groups

Author(s)

Ken Kelley (University of Notre Dame; KKelley@nd.edu)

References

Buyse, M., & Molenberghs, G. (1998). Criteria for the validation of surrogate endpoints in randomized experiments. *Biometrics, 54*, 1014–1029.

Ditlevsen, S., Christensen, U., Lynch, J., Damsgaard, M. T., & Keiding, N. (2005). The mediation proportion: A structural equation approach for estimating the proportion of exposure effect on outcome explained by an intermediate variable. *Epidemiology, 16*, 114–120.

Freedman, L. S. (2001). Confidence intervals and statistical power of the 'Validation' ratio for surrogate or intermediate endpoints. *Journal of Statistical Planning and Inference, 96*, 143–153.

Hansen, W. B., & McNeal, R. B. (1996). The law of maximum expected potential effect: Constraints placed on program effectiveness by mediator relationships. *Health Education Research, 11*, 501–507.

Huang, B., Sivaganesan, S., Succop, P., & Goodman, E. (2004). Statistical assessment of mediational effects for logistic mediational models. *Statistics in Medicine, 23*, 2713–2728.

Lachowicz, M. J., Preacher, K. J., & Kelley, K. (submitted). A novel measure of effect size for mediation analysis. Submited for publication.

Lindenberger, U., & Potter, U. (1998). The complex nature of unique and shared effects in hierarchical linear regression: Implications for developmental psychology. *Psychological Methods, 3*, 218–230.

MacKinnon, D. P. (2008). *Introduction to statistical mediation analysis*. Mahwah, NJ: Erlbaum.

Preacher, K. J., & Hayes, A. F. (2008b). Asymptotic and resampling strategies for assessing and comparing indirect effects in multiple mediator models. *Behavior Research Methods, 40*, 879–891.

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

Sobel, M. E. (1982). Asymptotic confidence intervals for indirect effects in structural equation models. In S. Leinhardt (Ed.), *Sociological Methodology 1982* (pp. 290–312). Washington DC: American Sociological Association.

Wen, Z., & Fan, X. (2015). Monotonicity of effect sizes: Questioning kappa-squared as mediation effect size measure. *Psychological Methods*, *20*, 193–203.

See Also

mediation.effect.plot, mediation.effect.bar.plot

Examples

```
## Not run:
############################################
# EXAMPLE 1
# Using the Jessor data discussed in Preacher and Kelley (2011), to illustrate
# the methods based on summary statistics.

mediation(S=rbind(c(2.26831107,  0.6615415, -0.08691755),
c(0.66154147,  2.2763549, -0.22593820), c(-0.08691755, -0.2259382,  0.09218055)),
N=432, x.location.S=1, mediator.location.S=2, dv.location.S=3, mean.x=7.157645,
mean.m=5.892785, mean.dv=1.649316, conf.level=.95)

############################################
# EXAMPLE 2
# Clear the workspace:
rm(list=ls(all=TRUE))

# An (unrealistic) example data (from Hayes)
Data <- rbind(
  c(-5.00, 25.00, -1.00),
  c(-4.00, 16.00, 2.00),
  c(-3.00, 9.00, 3.00),
  c(-2.00, 4.00, 4.00),
  c(-1.00, 1.00, 5.00),
  c(.00, .00, 6.00),
  c(1.00, 1.00, 7.00),
  c(2.00, 4.00, 8.00),
  c(3.00, 9.00, 9.00),
  c(4.00, 16.00, 10.00),
  c(5.00, 25.00, 13.00),
  c(-5.00, 25.00, -1.00),
  c(-4.00, 16.00, 2.00),
  c(-3.00, 9.00, 3.00),
  c(-2.00, 4.00, 4.00),
  c(-1.00, 1.00, 5.00),
  c(.00, .00, 6.00),
  c(1.00, 1.00, 7.00),
  c(2.00, 4.00, 8.00),
  c(3.00, 9.00, 9.00),
  c(4.00, 16.00, 10.00),
  c(5.00, 25.00, 13.00))


# Raw data example of the Hayes data.
mediation(x=Data[,1], mediator=Data[,2], dv=Data[,3], conf.level=.95)

# Sufficient statistics example of the Hayes data.
mediation(S=var(Data), N=22, x.location.S=1, mediator.location.S=2, dv.location.S=3,
```

```
mean.x=mean(Data[,1]), mean.m=mean(Data[,2]), mean.dv=mean(Data[,3]), conf.level=.95)

# Example had there been two groups.
gp.size <- length(Data[,1])/2 # adjust if using an odd number of observations.
grouping.variable <- c(rep(0, gp.size), rep(1, gp.size))
mediation(x=grouping.variable, mediator=Data[,2], dv=Data[,3])

############################################
# EXAMPLE 3
# Bootstrap of continuous data.
set.seed(12414) # Seed used for repeatability (there is nothing special about this seed)
bs.Results <- mediation(x=Data[,1], mediator=Data[,2], dv=Data[,3],
bootstrap=TRUE, B=5000, save.bs.replicates=TRUE)

ls() # Notice that Bootstrap.Replicates is available in the
workspace (if save.bs.replicates=TRUE in the above call).

#Now, given the Bootstrap.Replicates object, one can do whatever they want with them.

# See the names of the effect sizes (and their ordering)
colnames(Bootstrap.Replicates)

# Define IE as the indirect effect from the Bootstrap.Replicates object.
IE <- Bootstrap.Replicates$Indirect.Effect

# Summary statistics
mean(IE)
median(IE)
sqrt(var(IE))

# CIs from percentile perspective
quantile(IE, probs=c(.025, .975))

# Two-sided p-value.
## First, calculate obseved value of the indirect effect and extract it here.
IE.Observed <- mediation(x=Data[,1], mediator=Data[,2], dv=Data[,3],
conf.level=.95)$Effect.Sizes[1,]

## Now, find those values of the bootstrap indirect effects that are more extreme (in an absolute
## sense) than the indirect effect observed. Note that the p-value is 1 here because the observed
## indirect effect is exactly 0.
mean(abs(IE) >= abs(IE.Observed))

## End(Not run)
```

---

mediation.effect.bar.plot

Bar plots of mediation effects

Description

Provides an effect bar plot in the context of simple mediation.

Usage

```
mediation.effect.bar.plot(x, mediator, dv,
main = "Mediation Effect Bar Plot", width = 1, left.text.adj = 0,
right.text.adj = 0, rounding = 3, file = "", save.pdf = FALSE,
save.eps = FALSE, save.jpg = FALSE, ...)
```

Arguments

`x`	vector of the predictor/independent variable
`mediator`	vector of the mediator variable
`dv`	vector of the dependent/outcome variable
`main`	main title
`width`	width of bar, default 1
`left.text.adj`	for fine tuning left side text adjustment
`right.text.adj`	for fine tuning right side text adjustment
`rounding`	how to round so that the values displayed in the plot do not have too few or too many significant digits
`file`	file name of the plot to be saved (not necessary)
`save.pdf`	`TRUE` or `FALSE` if the produced figure should be saved as a PDF file
`save.eps`	`TRUE` or `FALSE` if the produced figure should be saved as an EPS file
`save.jpg`	`TRUE` or `FALSE` if the produced figure should be saved as a JPG file
`...`	optional additional specifications for nested functions

Details

Provides an effect bar for mediation (Bauer, Preacher, & Gil, 2006) may be used to plot the results of a mediation analysis compactly. Effect bars represent, in a single metric, the relative magnitudes of several values that are important for interpreting indirect effects. Preacher and Kelley (2011) discuss this plotting method also.

Value

Only a figure is returned

Author(s)

Ken Kelley (University of Notre Dame; KKelley@nd.edu)

References

Bauer, D. J., Preacher, K. J., & Gil, K. M. (2006). Conceptualizing and testing random indirect effects and moderated mediation in multilevel models: New procedures and recommendations. *Psychological Methods, 11*, 142–163.

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

See Also

`mediation`, `mediation.effect.bar.plot`

`mediation.effect.plot` *Visualizing mediation effects*

Description

Create a mediation effect plot

Usage

```
mediation.effect.plot(x, mediator, dv, ylab = "Dependent Variable",
xlab = "Mediator", main = "Mediation Effect Plot",
pct.from.top.a = 0.05, pct.from.left.c = 0.05, arrow.length.a = 0.05,
arrow.length.c = 0.05, legend.loc = "topleft", file = "", pch = 20,
xlim = NULL, ylim = NULL, save.pdf = FALSE, save.eps = FALSE,
save.jpg = FALSE, ...)
```

Arguments

`x`	vector of the predictor/independent variable
`mediator`	vector of the mediator variable
`dv`	vector of the dependent/outcome variable
`ylab`	*y*-axis title label
`xlab`	*x*-axis title label
`main`	main title label
`pct.from.top.a`	figure fine tuning adjustment
`pct.from.left.c`	figure fine tuning adjustment
`arrow.length.a`	figure fine tuning adjustment
`arrow.length.c`	figure fine tuning adjustment
`legend.loc`	specify the location of the legend
`file`	file name of the plot to be saved (not necessary)
`pch`	plotting character

`xlim`	limits for the *x*-axis
`ylim`	limits for the *y*-axis
`save.pdf`	`TRUE` or `FALSE` if the produced figure should be saved as a PDF file
`save.eps`	`TRUE` or `FALSE` if the produced figure should be saved as an EPS file
`save.jpg`	`TRUE` or `FALSE` if the produced figure should be saved as a JPG file
`...`	to incorporate options from interval functions

Details

Merrill (1994; see also MacKinnon, 2008; MacKinnon et al., 2007; Sy, 2004) presents a method that involves plotting the indirect effect as the vertical distance between two lines. Fritz and MacKinnon (2008) present a detailed exposition of this method too. Preacher and Kelley (2011) discuss this plotting method and implement their own code, which was also independently done as part of Fritz and MacKinnon (2008).

In this type of plot, the two horizontal lines correspond to the predicted values of Y regressed on X at the mean of X and at one unit above the mean of X. The distance between these two lines is thus $\hat{c}$. The two vertical lines correspond to predicted values of M regressed on X at the same two values of X. The distance between these lines is $\hat{a}$. The lines corresponding to the regression of Y on M (controlling for X) are plotted for the same two values of X.

Value

A figure is returned.

Note

Requires raw data.

Author(s)

Ken Kelley (University of Notre Dame; KKelley@nd.edu)

References

Fritz, M. S., & MacKinnon, D. P. (2008). A graphical representation of the mediated effect. *Behavior Research Methods, 40*, 55–60.

MacKinnon, D. P. (2008). *Introduction to statistical mediation analysis*. Mahwah, NJ: Erlbaum.

MacKinnon, D. P., Fairchild, A. J., & Fritz, M. S. (2007). Mediation analysis. *Annual Review of Psychology, 58*, 593–614.

Merrill, R. M. (1994). *Treatment effect evaluation in non-additive mediation models*. Unpublished dissertation, Arizona State University.

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

Sy, O. S. (2004). *Multilevel mediation analysis: Estimation and applications*. Unpublished dissertation, Kansas State University.

See Also

mediation.effect.plot, mediation.effect.bar.plot

---

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 (in press), 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=.49,
verbose=FALSE)
```

Arguments

`data`	the data for which to evalaute 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`	this is the 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 optimation 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 (University of Notre Dame; <kkelley@nd.edu>) and Bhargab Chattopadhyay (University of Texas - Dallas; <bhargab@utdallas.edu>)

References

Chattopadhyay, B., & Kelley, K. (in press). Estimation of the Coefficient of Variation with Minimum Risk: A Sequential Method for Minimizing Sampling Error and Study Cost. *Multivariate Behavioral Research*, *X*, X–X.

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.

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 (submitted, Psychological Methods), which considers study cost and accuracy of the estimated standardized mean difference simultaniously. This is important to specify that `mr.smd.R` 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`	is 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`	
`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 (submitted). See also Kelley and Rausch (2006) for additional information on the standardized mean difference.

Value

`Risk`	*Per group* sample size (this simply repeats what was supplied to the function)
`n1`	Sample size for group 1 (echos the input value)
`n1`	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 optimation 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 (University of Notre Dame; <`kkelley@nd.edu`>) and Bhargab Chattopadhyay (University of Texas - Dallas; <`bhargab@utdallas.edu`>)

References

Chattopadhyay, B., & Kelley, K. (submitted, minor revision requested). Estimating the standardized mean difference with minimum risk: Maximizing accuracy and minimizing cost with sequential estimation. *Psychological Methods*, *X*, X–X.

Chattopadhyay, B., & Kelley, K. (in press). Estimation of the Coefficient of Variation with Minimum Risk: A Sequential Method for Minimizing Sampling Error and Study Cost. *Multivariate Behavioral Research*, *X*, X–X.

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.

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 (submitted, minor revision requested)
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), 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)
```

---

power.density.equivalence.md

Density for power of two one-sided tests procedure (TOST) for equivalence

---

Description

A function to calculate density for the power of the two one-sided tests prodedure (TOST). (See package equivalence, function tost.)

Usage

```
power.density.equivalence.md(power_sigma, alpha = alpha, theta1 = theta1,
theta2 = theta2, diff = diff, sigma = sigma, n = n, nu = nu)
```

Arguments

`power_sigma`	x-value for integration
`alpha`	`alpha` level for the 2 *t*-tests (usually `alpha`=0.05). Confidence interval for full test is at level (1-2*α)
`theta1`	lower limit of equivalence interval on appropriate scale (regular or log)
`theta2`	upper limit of equivalence interval on appropriate scale
`diff`	true difference (ratio on log scale) in treatment means on appropriate scale
`sigma`	sqrt(error variance) as fraction (root MSE from ANOVA, or coefficient of variation)
`n`	number of subjects per treatment (number of total subjects for crossover design)
`nu`	degrees of freedom for sigma

Value

`power_density`	density at diff for power of TOST: the probability that the confidence interval will lie within ['theta1', 'theta2']

Author(s)

Kem Phillips; `<kemphillips@comcast.net>`

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*, No. 1, 1–8.

Phillips, K.F. (1990). Power of the Two One-Sided Tests Procedure in Bioquivalence, *Journal of Pharmacokinetics and Biopharmaceutics, 18*, No. 2, 139–144.

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*. 657–680.

See Also

`power.equivalence.md.plot`, `power.density.equivalence.md`

Examples

```
## Not run:
# This function is called by power.equivalence.md within
# the integrate function. It is integrated over
# appropriate limits to compute the power. Use

power.density.equivalence.md(.1, alpha=.05, theta1=-.2, theta2=.2, diff=.05,
sigma= .20, n=24, nu=22)

# The usage for the logarithmic scale is the same, except that
```

```
# theta1, theta2, and diff must be on that scale. That is, use log(.8), etc.

## End(Not run)
```

---

power.equivalence.md *Power of Two One-Sided Tests Procedure (TOST) for Equivalence*

Description

A function to calculate the power of the two one-sided tests prodedure (TOST). This is the probability that a confidence interval lies within a specified equivalence interval. (See also package `equivalence`, function `tost`.)

Usage

```
power.equivalence.md(alpha, logscale, ltheta1, ltheta2, ldiff, sigma, n, nu)
```

Arguments

`alpha`	*alpha* level for the 2 empht-tests (usually *alpha*=0.05). Confidence interval for full test is at level 1- 2*`alpha`
`logscale`	whether to use logarithmic scale (`TRUE`) or not (`FALSE`)
`ltheta1`	lower limit of equivalence interval
`ltheta2`	upper limit of equivalence interval
`ldiff`	true difference (ratio on log scale) in treatment means
`sigma`	`sqrt`(error variance) as fraction (root MSE from ANOVA, or coefficient of variation)
`n`	number of subjects per treatment (number of total subjects for crossover design)
`nu`	degrees of freedom for `sigma`

Value

`power`	Power of TOST; the probability that the confidence interval will lie within ['theta1', 'theta2'] given `sigma`, `n`, and `nu`

Author(s)

Kem Phillips; `<kemphillips@comcast.net>`

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*, No. 1, 1–8.

Phillips, K.F. (1990). Power of the Two One-Sided Tests Procedure in Bioquivalence, *Journal of Pharmacokinetics and Biopharmaceutics, 18*, No. 2, 139–144.

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*. 657–680.

Examples

```
 # Suppose that two formulations of a drug are to be compared on
 # the regular scale using a two-period crossover design, with
 # theta1 = -0.20, theta2 = 0.20, rm{CV} = 0.20, the
 # difference in the mean bioavailability is 0.05 (5 percent), and we choose
 # n=24, corresponding to 22 degrees of freedom.  We need to test
 # bioequivalence at the 5 percent significance level, which corresponds to
 # having a 90 percent confidence interval lying within (-0.20, 0.20). Then
 # the power will be 0.8029678.  This corresponds to Phillips (1990),
 # Table 1, 5th row, 5th column, and Figure 3.  Use

power.equivalence.md(.05, FALSE, -.2, .2, .05, .20, 24, 22)


# If the formulations are compared on the logarithmic scale with
# theta1 = 0.80, theta2 = 1.25, n=18 (16 degrees of freedom), and
# a ratio of test to reference of 1.05. Then the power will be 0.7922796.
# This corresponds to Diletti, Table 1, power=.80, CV=.20, ratio=1.05, and Figure 1c. Use

power.equivalence.md(.05, TRUE, .8, 1.25, 1.05, .20, 18, 16)
```

---

power.equivalence.md.plot

Plot power of Two One-Sided Tests Procedure (TOST) for Equivalence

---

Description

A function to plot the power of the two one-sided tests prodedure (TOST) for various alternatives. (See also package `equivalence`, function `tost`.)

Usage

```
power.equivalence.md.plot(alpha, logscale, theta1, theta2, sigma, n, nu, title2)
```

Arguments

`alpha`	*alpha* level for the 2 *t*-tests (usually *alpha*=0.05). Confidence interval for full test is at level 1- 2*`alpha`
`logscale`	whether to use logarithmic scale `TRUE` or not `FALSE`
`theta1`	lower limit of equivalence interval
`theta2`	upper limit of equivalence interval
`sigma`	`sqrt`(error variance) as fraction (root MSE from ANOVA, or coefficient of variation)
`n`	number of subjects per treatment (number of total subjects for crossover design)
`nu`	degrees of freedom for `sigma`
`title2`	Title appearing at bottom of plot

Value

`power`	Plot of power of TOST (probability that (1-2*`alpha`) confidence interval will lie within (`theta1`, `theta2`) given `sigma`, `n`, and `nu`. Also returns matrix of 201 differences between `theta1` and `theta2` as first column, and power values corresponding to `n` for other columns.

Author(s)

Kem Phillips; <kemphillips@comcast.net>

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*, No. 1, 1-8.

Phillips, K.F. (1990) Power of the Two One-Sided Tests Procedure in Bioquivalence, *Journal of Pharmacokinetics and Biopharmaceutics, 18*, No. 2, 139-144.

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*. 657-680.

Examples

```
## Not run:
# Suppose that two formulations of a drug are to be compared
# on the regular scale using a two-period crossover design,
# with theta1 = -0.20, theta2 = 0.20, rm(CV) = 0.20, and
# we choose
n<-c(9,12,18,24,30,40,60)

# corresponding to
nu<-c(7,10,16,22,28,38,58)

# degrees of freedom.  We need to test bioequivalence at the
```

```
# .05 significance level, which corresponds to having a .90 confidence
# interval lying within (-0.20, 0.20). This corresponds to
# Phillips (1990),  Figure 3.  Use

power.equivalence.md.plot(.05, FALSE, -.2, .2, .20, n, nu, 'Phillips Figure 3')

# If the formulations are compared on the logarithmic scale with
# theta1 = 0.80, theta2 = 1.25, and

n<-c(8,12,18,24,30,40,60)

# corresponding  to
nu<-c(6,10,16,22,28,38,58)

# degrees of freedom. This corresponds to Diletti, Figure 1c. Use

power.equivalence.md.plot(.05, TRUE, .8, 1.25, .20, n, nu, 'Diletti, Figure 1c')

## End(Not run)
```

`prof.salary` *Cohen et. al. (2003)'s professor salary data set*

Description

The data set of the salaries and other information of 62 some professors in Cohen et. al. (2003, pp. 81-82).

Usage

```
data(prof.salary)
```

Format

A data frame with 62 observations on the following 6 variables.

`id` the identification number

`time` the time since getting the Ph.D. degree

`pub` the number of publications

`sex` the gender, 1 for female and 0 for male

`citation` the citation count

`salary` the professor's current salary

Source

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (3rd ed.). Mahwah, NJ: Erlbaum.

References

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (3rd ed.). Mahwah, NJ: Erlbaum.

Examples

```
data(prof.salary)
```

`s.u` *Unbiased estimate of the population standard deviation*

Description

Transforms the usual (and biased) estimate of the standard deviation into an unbiased estimator.

Usage

```
s.u(s=NULL, N=NULL, X=NULL)
```

Arguments

`s`	the usual estimate of the standard deviation (i.e., the square root of the unibased estimate of the variance)
`N`	sample size `s` is based
`X`	vector of scores in which the unbiased estimate of the standard deviation should be calculated

Details

Returns the unbiased estimate for the standard deviation.

Value

The unbiased estimate for the standard deviation.

Author(s)

Ken Kelley (University of Notre Dame; `<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.

Examples

```
set.seed(113)
X <- rnorm(10, 100, 15)

# Square root of the unbiased estimate of the variance (not unbiased)
var(X)^.5

# One way to implement the function.
s.u(s=var(X)^.5, N=length(X))

# Another way to implement the function.
s.u(X=X)
```

`Sigma.2.SigmaStar` *Construct a covariance matrix with specified error of approximation*

Description

This function implements Cudeck & Browne's (1992) method to construct a covariance matrix in the structural equation modeling (SEM) context. Given an SEM model and its model parameters, a covariance matrix is obtained so that (a) the population discrepancy due to approximation equals a certain specified value; and (b) the population model parameter vector is the minimizer of the discrepancy function.

Usage

```
Sigma.2.SigmaStar(model, model.par, latent.var, discrep, ML = TRUE)
```

Arguments

`model`	an RAM (reticular action model; e.g., McArdle & McDonald, 1984) specification of a structural equation model, and should be of class `mod`. The model is specified in the same manner as does the `sem` package; see `sem` and `specify.model` for detailed documentations about model specifications in the RAM notation.
`model.par`	a vector containing the model parameters. The names of the elements in `theta` must be the same as the names of the model parameters specified in `model`.
`latent.var`	a vector containing the names of the latent variables
`discrep`	the desired discrepancy function minimum value
`ML`	the discrepancy function to be used, if `ML=TRUE` then the discrepancy function is based on normal theory maximum likelihood

Details

This function constructs a covariance matrix Σ^* such that $\Sigma^* = \Sigma(\theta) + E$, where $\Sigma(\theta)$ is the population model-implied covariance matrix, and E is a matrix containing the errors due to approximation. The matrix E is chosen so that the discrepancy function $F(\Sigma^*, \Sigma(\theta))$ has the specified discrepancy value.

This function uses the same notation to specify SEM models as does `sem`. Please refer to `sem` for more detailed documentation about model specification and the RAM notation. For technical discussion on how to obtain the model implied covariance matrix in the RAM notation given model parameters, see McArdle and McDonald (1984).

Value

`Sigma.star`	the population covariance matrix of manifest variables
`Sigma_theta`	the population model-implied covariance matrix
`E`	the matrix containing the population errors of approximation, i.e., `Sigma.star - Sigma_theta`

Author(s)

Keke Lai (University of California-Merced)

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.

Fox, J. (2006). Structural equation modeling with the sem package in R. *Structural Equation Modeling, 13*, 465–486.

McArdle, J. J., & McDonald, R. P. (1984). Some algebraic properties of the reticular action model. *British Journal of Mathematical and Statistical Psychology, 37*, 234–251.

See Also

`sem`; `specify.model`; `theta.2.Sigma.theta`

Examples

```
## Not run:
library(sem)

###############
## EXAMPLE 1; a CFA model with three latent variables and nine indicators.
###############

# To specify the model
model.cfa<-specify.model()
xi1 -> x1, lambda1, 0.6
xi1 -> x2, lambda2, 0.7
xi1 -> x3, lambda3, 0.8
xi2 -> x4, lambda4, 0.65
```

```
xi2 -> x5, lambda5, 0.75
xi2 -> x6, lambda6, 0.85
xi3 -> x7, lambda7, 0.5
xi3 -> x8, lambda8, 0.7
xi3 -> x9, lambda9, 0.9
xi1 <-> xi1, NA, 1
xi2 <-> xi2, NA, 1
xi3 <-> xi3, NA, 1
xi1 <-> xi2, phi21, 0.5
xi1 <-> xi3, phi31, 0.4
xi2 <-> xi3, phi32, 0.6
x1 <-> x1, delta11, 0.36
x2 <-> x2, delta22, 0.5
x3 <-> x3, delta33, 0.9
x4 <-> x4, delta44, 0.4
x5 <-> x5, delta55, 0.5
x6 <-> x6, delta66, 0.6
x7 <-> x7, delta77, 0.6
x8 <-> x8, delta88, 0.7
x9 <-> x9, delta99, 0.7


# To specify model parameters
theta <- c(0.6, 0.7, 0.8,
0.65, 0.75, 0.85,
0.5, 0.7, 0.9,
0.5, 0.4, 0.6,
0.8, 0.6, 0.5,
0.6, 0.5, 0.4,
0.7, 0.7, 0.6)

names(theta) <- c("lambda1", "lambda2", "lambda3",
"lambda4","lambda5", "lambda6",
"lambda7", "lambda8","lambda9",
"phi21", "phi31", "phi32",
"delta11", "delta22","delta33",
"delta44", "delta55","delta66",
"delta77", "delta88","delta99")

res.matrix <- Sigma.2.SigmaStar(model=model.cfa, model.par=theta,
latent.var=c("xi1", "xi2", "xi3"), discrep=0.06)

# res.matrix

# To verify the returned covariance matrix; the model chi-square
# should be equal to (N-1) times the specified discrepancy value.
# Also the "point estimates" of model parameters should be
# equal to the specified model parameters

# res.sem<-sem(model.cfa, res.matrix$Sigma.star, 1001)
# summary(res.sem)

# To construct a covariance matrix so that the model has
```

```
# a desired population RMSEA value, one can transform the RMSEA
# value to the discrepancy value

res.matrix <- Sigma.2.SigmaStar(model=model.cfa, model.par=theta,
latent.var=c("xi1", "xi2", "xi3"), discrep=0.075*0.075*24)

# To verify the population RMSEA value
# res.sem<-sem(model.cfa, res.matrix$Sigma.star, 1000000)
# summary(res.sem)

###############
## EXAMPLE 2; an SEM model with five latent variables
###############

model.5f <- specify.model()
eta1 -> y4, NA, 1
eta1 -> y5, lambda5, NA
eta2 -> y1, NA, 1
eta2 -> y2, lambda2, NA
eta2 -> y3, lambda3, NA
xi1 -> x1, NA, 1
xi1 -> x2, lambda6, NA
xi1 -> x3, lambda7, NA
xi2 -> x4, NA, 1
xi2 -> x5, lambda8, NA
xi3 -> x6, NA, 1
xi3 -> x7, lambda9, NA
xi3 -> x8, lambda10, NA
xi1 -> eta1, gamma11, NA
xi2 -> eta1, gamma12, NA
xi3 -> eta1, gamma13, NA
xi3 -> eta2, gamma23, NA
eta1 -> eta2, beta21, NA
xi1 <-> xi2, phi21, NA
xi1 <-> xi3, phi31, NA
xi3 <-> xi2, phi32, NA
xi1 <-> xi1, phi11, NA
xi2 <-> xi2, phi22, NA
xi3 <-> xi3, phi33, NA
eta1 <-> eta1, psi11, NA
eta2 <-> eta2, psi22, NA
y1 <-> y1, eplison11, NA
y2 <-> y2, eplison22, NA
y3 <-> y3, eplison33, NA
y4 <-> y4, eplison44, NA
y5 <-> y5, eplison55, NA
x1 <-> x1, delta11, NA
x2 <-> x2, delta22, NA
x3 <-> x3, delta33, NA
x4 <-> x4, delta44, NA
x5 <-> x5, delta55, NA
x6 <-> x6, delta66, NA
x7 <-> x7, delta77, NA
```

```
x8 <-> x8, delta88, NA


theta <- c(0.84, 0.8, 0.9,
1.26, 0.75, 1.43, 1.58, 0.83,
0.4, 0.98, 0.52, 0.6,0.47,
0.12, 0.14, 0.07,
0.44, 0.22, 0.25,
0.3, 0.47,
0.37, 0.5, 0.4, 0.4, 0.58,
0.56,0.3, 0.6, 0.77, 0.54, 0.75, 0.37, 0.6)

names(theta) <- c(
"lambda5","lambda2","lambda3",
"lambda6","lambda7","lambda8","lambda9","lambda10" ,
"gamma11",  "gamma12","gamma13" ,  "gamma23" ,  "beta21",
"phi21","phi31", "phi32",
"phi11","phi22",  "phi33",
"psi11" ,    "psi22"   ,
"eplison11","eplison22" ,"eplison33", "eplison44" ,"eplison55",
  "delta11"  , "delta22" ,  "delta33" ,  "delta44" ,  "delta55" ,  "delta66",
"delta77" , "delta88")

# To construct a covariance matrix so that the model has
# a population RMSEA of 0.08

res.matrix <- Sigma.2.SigmaStar(model=model.5f, model.par=theta,
latent.var=c("xi1", "xi2", "xi3", "eta1","eta2"), discrep=0.08*0.08*57)

# To verify
# res.sem<- sem(model.5f, res.matrix$Sigma.star, 1000000)
# summary(res.sem)

## End(Not run)
```

`signal.to.noise.R2` *Signal to noise using squared multiple correlation coefficient*

Description

Function that calculates five different signal-to-noise ratios using the squared multiple correlation coefficient.

Usage

```
signal.to.noise.R2(R.Square, N, p)
```

Arguments

R.Square	usual estimate of the squared multiple correlation coefficient (with no adjustments)
N	sample size
p	number of predictors

Details

The method of choice is phi2.UMVUE.NL, but it requires p of 5 or more. In situations where p < 5, it is suggested that phi2.UMVUE.L be used.

Value

phi2.hat	Basic estimate of the signal-to-noise ratio using the usual estimate of the squared multiple correlation coefficient: phi2.hat=R.Square/(1-R.Square)
phi2.adj.hat	Estimate of the signal-to-noise ratio using the usual adjusted R Square in place of *R*-Square: phi2.hat=*Adj.R2*/(1-*Adj.R2*)
phi2.UMVUE	Muirhead's (1985) unique minimum variance unbiased estimate of the signal-to-noise ratio (Muirhead uses theta-U): see reference or code for formula
phi2.UMVUE.L	Muirhead's (1985) unique minimum variance unbiased linear estimate of the signal-to-noise ratio (Muirhead uses theta-L): see reference or code for formula
phi2.UMVUE.NL	Muirhead's (1985) unique minimum variance unbiased nonlinear estimate of the signal-to-noise ratio (Muirhead uses theta-NL); requires the number of predictors to be greater than five: see reference or code for formula

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>)

References

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

Muirhead, R. J. (1985). Estimating a particular function of the multiple correlation coefficient. *Journal of the American Statistical Association, 80*, 923–925.

See Also

ci.R2, ss.aipe.R2

Examples

```
signal.to.noise.R2(R.Square=.5, N=50, p=2)
signal.to.noise.R2(R.Square=.5, N=50, p=5)
signal.to.noise.R2(R.Square=.5, N=100, p=2)
signal.to.noise.R2(R.Square=.5, N=100, p=5)
```

`smd` *Standardized mean difference*

Description

Function to calculate the standardized mean difference (regular or unbiased) using either raw data or summary measures.

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

Returns the estimated standardized mean difference.

Author(s)

Ken Kelley (University of Notre Dame; <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*, 532–574.

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 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.

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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`, `conf.limits.nct`, `ss.aipe`

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

Function to calculate the standardized mean difference (regular or unbiased) using the control group standard deviation as the basis of standardization (for either raw data or summary measures).

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

Returns the estimated standardized mean difference using the control group standard deviation as the basis of standardization.

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`)

References

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 2*, 107–128.

Glass, G. (1976). Primary, secondary, and meta-analysis of research. *Educational Researcher, 5*, 3–8.

See Also

`smd`, `conf.limits.nct`

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=sd.c)
smd.c(Mean.T=M.T, Mean.C=M.C, s=sd.c, n.C=n.c, Unbiased=TRUE)
```

---

ss.aipe.c — *Sample size planning for an ANOVA contrast from the Accuracy in Parameter Estimation (AIPE) perspective*

Description

A function to calculate the appropriate sample size *per group* for the (unstandardized) ANOVA contrast so that the width of the confidence interval is sufficiently narrow.

Usage

```
ss.aipe.c(error.variance = NULL, c.weights, width, conf.level = 0.95,
assurance = NULL, certainty = 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)

`certainty` an alias for `assurance`

`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

`n` the necessary sample size *per group*

Note

Be sure to use the error varaince and not its square root (i.e., the standard deviation of the errors).

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>), Keke Lai

References

Kelley, K., Maxwell, S. E., & Rausch, J. R. (2003). Obtaining power or obtaining precesion: Delineating methods of sample size planning. *Evaluation and the Health Professions, 26*, 258–287.

Maxwell, S. E., & Delaney, H. D. (2004). *Designing experiments and analyzing data: A model comparison perspective*. Mahwah, NJ: Erlbaum.

See Also

`ss.aipe.sc`, `ss.aipe.c.ancova`, `ci.c`

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

A function to calculate the appropriate sample size per group for the (unstandardized) contrast, in one-covariate randomized ANCOVA, so that the width of the confidence interval is sufficiently narrow.

Usage

```
ss.aipe.c.ancova(error.var.ancova = NULL, error.var.anova = NULL,
rho = NULL, c.weights, width, conf.level = 0.95,
assurance = NULL, certainty = 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)
- `certainty` an alias for `assurance`

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

- `n` the necessary sample size *per group*

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`); Keke Lai <Lai.15@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*, 258-287.

Maxwell, S. E., & Delaney, H. D. (2004). *Designing experiments and analyzing data: A model comparison perspective*. Mahwah, NJ: Erlbaum.

See Also

`ci.c.ancova`, `ci.sc.ancova`, `ss.aipe.c`

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, certainty=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)

`certainty` an alias for `assurance`

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 `mu.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

`Psi.obs` the observed (unstandardized) contrast

`se.Psi` the standard error of the observed (unstandardized) contrast

`se.Psi.restricted`
: the standard error of the observed (unstandardized) contrast calculated by ignoring the covariate

`se.res.over.se.full`
: the ratio of contrast's full standard error over the restricted one in each iteration

`width.obs` full confidence interval width

`Type.I.Error` Type I error happens in each iteration

`Type.I.Error.Upper`
: Type I error happens in the upper end in each iteration

`Type.I.Error.Lower`
: Type I error happens in the lower end in each iteration

`Type.I.Error` percentage of Type I error happened in the entire simulation

`Type.I.Error.Upper` percentage of Type I error happened in the upper end in the entire simulation

`Type.I.Error.Lower` percentage of Type I error happened in the lower end in the entire simulation

`width.NARROWER.than.desired` percentage of obtained widths that are narrower than the desired width

`Mean.width.obs` mean width of the obtained full confidence intervals

`Median.width.obs` median width of the obtained full confidence intervals

`Mean.se.res.vs.se.full` the mean of the ratios of contrast's full standard error over the restricted one

`Psi.pop` population (unstandardized) contrast

`Contrast.Weights` contrast weights

`mu.y` the response's population mean of each group

`mu.x` the population mean of the covariate

`sigma.x` the population standard deviation of the covariate

`Sample.Size.per.Group` sample size per group

`conf.level` the desired confidence interval coverage, (i.e., 1 - Type I error rate)

`assurance` specified `assurance`

`rho` population correlation coefficient of the response and the covariate

`est.rho` estimated correlation coefficient of the response and the covariate

`true.error.var.ANOVA` population error variance of the ANOVA model

`est.error.var.ANOVA` estimated error variance of the ANOVA model

Author(s)

Keke Lai (University of Notre Dame; `<Lai.15@ND.Edu>`)

Examples

```
## Not run:
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=1000, 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=1000,
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)

## End(Not run)
```

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.nclus.fixedwidth(width, nindiv, prtreat, tauy=NULL, sigma2y=NULL,
totalvar=NULL, iccy=NULL, r2between = 0, r2within = 0, numpredictor = 0,
assurance=NULL, conf.level = 0.95, cluscost=NULL, indivcost=NULL, diffsize=NULL)
ss.aipe.crd.nindiv.fixedwidth(width, nclus, prtreat, tauy=NULL, sigma2y=NULL,
totalvar=NULL, iccy=NULL, r2between = 0, r2within = 0, numpredictor = 0,
assurance=NULL, conf.level = 0.95, cluscost=NULL, indivcost=NULL, diffsize=NULL)
ss.aipe.crd.nclus.fixedbudget(budget, nindiv, cluscost = 0, indivcost = 1,
prtreat = NULL, tauy=NULL, sigma2y=NULL, totalvar=NULL, iccy=NULL, r2between = 0,
r2within = 0, numpredictor = 0, assurance=NULL, conf.level = 0.95, diffsize=NULL)
ss.aipe.crd.nindiv.fixedbudget(budget, nclus, cluscost = 0, indivcost = 1,
prtreat = NULL, tauy=NULL, sigma2y=NULL, totalvar=NULL, iccy=NULL, r2between = 0,
r2within = 0, numpredictor = 0, assurance=NULL, conf.level = 0.95, diffsize=NULL)
ss.aipe.crd.both.fixedbudget(budget, cluscost=0, indivcost=1, prtreat, tauy=NULL,
sigma2y=NULL, totalvar=NULL, iccy=NULL, r2between = 0, r2within = 0,
numpredictor = 0, assurance=NULL, conf.level = 0.95, diffsize=NULL)
ss.aipe.crd.both.fixedwidth(width, cluscost=0, indivcost=1, prtreat, tauy=NULL,
sigma2y=NULL, totalvar=NULL, iccy=NULL, r2between = 0, r2within = 0,
numpredictor = 0, assurance=NULL, conf.level = 0.95, diffsize=NULL)
```

Arguments

width	The desired width of the confidence interval of the unstandardized means difference
budget	The desired amount of budget
nclus	The desired number of clusters
nindiv	The number of individuals in each cluster (cluster size)
prtreat	The proportion of treatment clusters
cluscost	The cost of collecting a new cluster regardless of the number of individuals collected in each cluster
indivcost	The cost of collecting a new individual
tauy	The residual variance in the between level before accounting for the covariate
sigma2y	The residual variance in the within level before accounting for the covariate

`totalvar`	The total resiudal variance before accounting for the covariate
`iccy`	The intraclass correlation of the dependent variable
`r2within`	The proportion of variance explained in the within level (used when `covariate = TRUE`)
`r2between`	The proportion of variance explained in the between level (used when `covariate = TRUE`)
`numpredictor`	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
`diffsize`	Difference cluster size specification. The difference in cluster sizes can be specified in two ways. First, users may specify cluster size as integers, which can be negative or positive. The resulting cluster sizes will be based on the estimated cluster size adding by the specified vectors. For example, if the cluster size is 25, the number of clusters is 10, and the specified different cluster size is `c(-1, 0, 1)`, the cluster sizes will be 24, 25, 26, 24, 25, 26, 24, 25, 26, and 24. Second, users may specify cluster size as positive decimals. The resulting cluster size will be based on the estimated cluster size multiplied by the specified vectors. For example, if the cluster size is 25, the number of clusters is 10, and the specified different cluster size is `c(-1, 0, 1)`, the cluster sizes will be 24, 25, 26, 24, 25, 26, 24, 25, 26, and 24. If `NULL`, the cluster size is equal across clusters.

Details

Here are the functions' descriptions:

- `ss.aipe.crd.nclus.fixedwidth` Find the number of clusters given a specified width of the confidence interval and the cluster size
- `ss.aipe.crd.nindiv.fixedwidth` Find the cluster size given a specified width of the confidence interval and the number of clusters
- `ss.aipe.crd.nclus.fixedbudget` Find the number of clusters given a budget and the cluster size
- `ss.aipe.crd.nindiv.fixedbudget` Find the cluster size given a budget and the number of clusters
- `ss.aipe.crd.both.fixedbudget` 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.fixedwidth` 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.nclus.fixedwidth` and `ss.aipe.crd.nclus.fixedbudget` functions provide the number of clusters. The `ss.aipe.crd.nindiv.fixedwidth` and `ss.aipe.crd.nindiv.fixedbudget` functions provide the cluster size. The `ss.aipe.crd.both.fixedbudget` and `ss.aipe.crd.both.fixedwidth` provide the number of clusters and the cluster size, respectively.

Author(s)

Sunthud Pornprasertmanit (<psunthud@gmail.com>)

References

Pornprasertmanic, S., & Schneider, W. J. (2014). Accuracy in parameter estimation in cluster randomized designs. *Psychological Methods*, *19*, 356–379.

Examples

```
## Not run:
# Examples for each function
ss.aipe.crd.nclus.fixedwidth(width=0.3, nindiv=30, prtreat=0.5, tauy=0.25, sigma2y=0.75)

ss.aipe.crd.nindiv.fixedwidth(width=0.3, nclus=250, prtreat=0.5, tauy=0.25, sigma2y=0.75)

ss.aipe.crd.nclus.fixedbudget(budget=10000, nindiv=20, cluscost=20, indivcost=1)

ss.aipe.crd.nindiv.fixedbudget(budget=10000, nclus=30, cluscost=20, indivcost=1,
prtreat=0.5, tauy=0.05, sigma2y=0.95, assurance=0.8)

ss.aipe.crd.both.fixedbudget(budget=10000, cluscost=30, indivcost=1, prtreat=0.5, tauy=0.25,
sigma2y=0.75)

ss.aipe.crd.both.fixedwidth(width=0.3, cluscost=0, indivcost=1, prtreat=0.5, tauy=0.25,
sigma2y=0.75)

# Examples for different cluster size
ss.aipe.crd.nclus.fixedwidth(width=0.3, nindiv=30, prtreat=0.5, tauy=0.25, sigma2y=0.75,
diffsize = c(-2, 1, 0, 2, -1, 3, -3, 0, 0))

ss.aipe.crd.nclus.fixedwidth(width=0.3, nindiv=30, prtreat=0.5, tauy=0.25, sigma2y=0.75,
diffsize = c(0.6, 1.2, 0.8, 1.4, 1, 1, 1.1, 0.9))

## End(Not run)
```

`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 (2010, submitted).

Usage

```
ss.aipe.crd.es.nclus.fixedwidth(width, nindiv, es, estype = 1, iccy, prtreat,
r2between = 0, r2within = 0, numpredictor = 0, assurance=NULL,
conf.level = 0.95, nrep = 1000, iccz = NULL, seed = 123321, multicore = FALSE,
numProc=NULL, cluscost=NULL, indivcost=NULL, diffsize=NULL)
ss.aipe.crd.es.nindiv.fixedwidth(width, nclus, es, estype = 1, iccy, prtreat,
r2between = 0, r2within = 0, numpredictor = 0, assurance=NULL,
conf.level = 0.95, nrep = 1000, iccz = NULL, seed = 123321, multicore = FALSE,
numProc=NULL, cluscost=NULL, indivcost=NULL, diffsize=NULL)
ss.aipe.crd.es.nclus.fixedbudget(budget, nindiv, cluscost, indivcost, nrep=NULL,
prtreat=NULL, iccy=NULL, es=NULL, estype = 1, numpredictor = 0,
iccz=NULL, r2within=NULL, r2between=NULL, assurance=NULL,
seed=123321, multicore=FALSE, numProc=NULL, conf.level=0.95, diffsize=NULL)
ss.aipe.crd.es.nindiv.fixedbudget(budget, nclus, cluscost, indivcost, nrep=NULL,
prtreat=NULL, iccy=NULL, es=NULL, estype = 1, numpredictor = 0,
iccz=NULL, r2within=NULL, r2between=NULL, assurance=NULL,
seed=123321, multicore=FALSE, numProc=NULL, conf.level=0.95, diffsize=NULL)
ss.aipe.crd.es.both.fixedbudget(budget, cluscost=0, indivcost=1, es, estype = 1,
iccy, prtreat, r2between = 0, r2within = 0, numpredictor = 0, assurance=NULL,
conf.level = 0.95, nrep = 1000, iccz = NULL, seed = 123321, multicore = FALSE,
numProc=NULL, diffsize=NULL)
ss.aipe.crd.es.both.fixedwidth(width, cluscost=0, indivcost=1, es, estype = 1,
iccy, prtreat, r2between = 0, r2within = 0, numpredictor = 0, assurance=NULL,
conf.level = 0.95, nrep = 1000, iccz = NULL, seed = 123321, multicore = FALSE,
numProc=NULL, diffsize=NULL)
```

Arguments

`width`	The desired width of the confidence interval of the unstandardized means difference
`budget`	The desired amount of budget
`nclus`	The desired number of clusters
`nindiv`	The number of individuals in each cluster (cluster size)
`prtreat`	The proportion of treatment clusters
`cluscost`	The cost of collecting a new cluster regardless of the number of individuals collected in each cluster
`indivcost`	The cost of collecting a new individual
`iccy`	The intraclass correlation of the dependent variable
`es`	The amount of effect size
`estype`	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)
`numpredictor`	If 1, a single covariate is included into the model. If 0, the no-covariate model is used. This function cannot handle multiple covariates. Therefore, only the values of 0 and 1 are allowed.

`iccz`	The intraclass correlation of the covariate (used when `covariate = TRUE`). If `iccz = 0`, the within-level covariate will be only used. If `iccz = 1`, the between-level covariate will be only used.
`r2within`	The proportion of variance explained in the within level (used when `covariate = TRUE`)
`r2between`	The proportion of variance explained in the between level (used when `covariate = TRUE`)
`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)
`nrep`	The number of replications used in a priori Monte Carlo simulation
`seed`	A desired seed number
`multicore`	Use multiple processors within a computer. Specify as `TRUE` to use it.
`numProc`	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.
`conf.level`	The desired level of confidence for the confidence interval
`diffsize`	Difference cluster size specification. The difference in cluster sizes can be specified in two ways. First, users may specify cluster size as integers, which can be negative or positive. The resulting cluster sizes will be based on the estimated cluster size adding by the specified vectors. For example, if the cluster size is 25, the number of clusters is 10, and the specified different cluster size is `c(-1, 0, 1)`, the cluster sizes will be 24, 25, 26, 24, 25, 26, 24, 25, 26, and 24. Second, users may specify cluster size as positive decimals. The resulting cluster size will be based on the estimated cluster size multiplied by the specified vectors. For example, if the cluster size is 25, the number of clusters is 10, and the specified different cluster size is `c(-1, 0, 1)`, the cluster sizes will be 24, 25, 26, 24, 25, 26, 24, 25, 26, and 24. If `NULL`, the cluster size is equal across clusters.

Details

Here are the functions' descriptions:

- `ss.aipe.crd.es.nclus.fixedwidth` Find the number of clusters given a specified width of the confidence interval and the cluster size
- `ss.aipe.crd.es.nindiv.fixedwidth` Find the cluster size given a specified width of the confidence interval and the number of clusters
- `ss.aipe.crd.es.nclus.fixedbudget` Find the number of clusters given a budget and the cluster size
- `ss.aipe.crd.es.nindiv.fixedbudget` Find the cluster size given a budget and the number of clusters
- `ss.aipe.crd.es.both.fixedbudget` 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.fixedwidth` 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.es.nclus.fixedwidth` and `ss.aipe.crd.es.nclus.fixedbudget` functions provide the number of clusters. The `ss.aipe.crd.es.nindiv.fixedwidth` and `ss.aipe.crd.es.nindiv.fixedbudget` functions provide the cluster size. The `ss.aipe.crd.es.both.fixedbudget` and `ss.aipe.crd.es.both.fixedwidth` provide the number of clusters and the cluster size, respectively.

Author(s)

Sunthud Pornprasertmanit (<psunthud@gmail.com>)

References

Boker, S., M., N., Maes, H., Wilde, M., Spiegel, M., Brick, T., et al. (2011). OpenMx: An open source extended structural equation modeling framework. *Psychometrika, 76,* 306-317.

Cheung, M. W.-L. (2009). Constructing approximate confidence intervals for parameters with structural constructing approximate confidence intervals for parameters with structural equation models. *Structural Equation Modeling, 16,* 267-294.

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.

Pornprasertmanic, S., & Schneider, W. J. (2014). Accuracy in parameter estimation in cluster randomized designs. *Psychological Methods*, *19*, 356–379.

Examples

```
## Not run:
# Examples for each function
ss.aipe.crd.es.nclus.fixedwidth(width=0.3, nindiv=20, es=0.5, estype=1, iccy=0.25, prtreat=0.5,
nrep=20)

ss.aipe.crd.es.nindiv.fixedwidth(width=0.3, 250, es=0.5, estype=1, iccy=0.25, prtreat=0.5,
nrep=20)

ss.aipe.crd.es.nclus.fixedbudget(budget=1000, nindiv=20, cluscost=0, indivcost=1, nrep=20,
prtreat=0.5, iccy=0.25, es=0.5)

ss.aipe.crd.es.nindiv.fixedbudget(budget=1000, nclus=200, cluscost=0, indivcost=1, nrep=20,
prtreat=0.5, iccy=0.25, es=0.5)

ss.aipe.crd.es.both.fixedbudget(budget=1000, cluscost=5, indivcost=1, es=0.5, estype=1,
iccy=0.25, prtreat=0.5, nrep=20)

ss.aipe.crd.es.both.fixedwidth(width=0.5, cluscost=5, indivcost=1, es=0.5, estype=1, iccy=0.25,
prtreat=0.5, nrep=20)

# Examples for different cluster size
ss.aipe.crd.es.nclus.fixedwidth(width=0.3, nindiv=20, es=0.5, estype=1, iccy=0.25, prtreat=0.5,
nrep=20, diffsize = c(-2, 1, 0, 2, -1, 3, -3, 0, 0))

ss.aipe.crd.es.nclus.fixedwidth(width=0.3, nindiv=20, es=0.5, estype=1, iccy=0.25, prtreat=0.5,
```

```
nrep=20, diffsize = c(0.6, 1.2, 0.8, 1.4, 1, 1, 1.1, 0.9))

## End(Not run)
```

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.

Usage

```
ss.aipe.cv(C.of.V = NULL, width = NULL, conf.level = 0.95,
degree.of.certainty = NULL, assurance=NULL, certainty=NULL,
mu = NULL, sigma = NULL, alpha.lower = NULL, alpha.upper = NULL,
Suppress.Statement = TRUE, sup.int.warns = TRUE, ...)
```

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
degree.of.certainty	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)
assurance	an alias for degree.of.certainty
certainty	an alias for degree.of.certainty
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
Suppress.Statement	Suppress a message restating the input specifications
sup.int.warns	suppress internal function warnings (e.g., warnings associated with qt)
...	for modifying parameters of functions this function calls

Value

Returns the necessary sample size given the input specifications.

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>)

See Also

ss.aipe.cv.sensitivity, cv

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 .25.
ss.aipe.cv(C.of.V=.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, degree.of.certainty=.99)
```

---

ss.aipe.cv.sensitivity

Sensitivity analysis for sample size planning given the Accuracy in Parameter Estimation approach for the coefficient of variation.

Description

Performs sensitivity analysis for sample size determination for the coefficient of variation given a population coefficient of variation (or population mean and standard deviation) and goals for the sample size procedure. Allows one to determine the effect of being wrong when estimating the population coefficient of variation in terms of the width of the obtained (two-sided) confidence intervals.

Usage

```
ss.aipe.cv.sensitivity(True.C.of.V = NULL, Estimated.C.of.V = NULL,
width = NULL, degree.of.certainty = NULL, assurance=NULL, certainty=NULL,
mean = 100, Specified.N = NULL, conf.level = 0.95,
G = 1000, print.iter = TRUE)
```

Arguments

True.C.of.V	population coefficient of variation
Estimated.C.of.V	estimated coefficient of variation
width	desired confidence interval width
degree.of.certainty	parameter to ensure confidence interval width with a specified degree of certainty (must be NULL or between zero and unity)
assurance	the alias for degree.of.certainty

`certainty`	an alias for `degree.of.certainty`
`mean`	Some arbitrary value that the simulation uses to generate data (the variance of the data is determined by the mean and the coefficient of variation)
`Specified.N`	selected sample size to use in order to determine distributional properties of at a given value of sample size (not used with `Estimated.C.of.V`)
`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

Details

For sensitivity analysis when planning sample size given the desire to obtain narrow confidence intervals for the population coefficient of variation. Given a population value and an estimated value, one can determine the effects of incorrectly specifying the population coefficient of variation (`True.C.of.V`) 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 `degree.of.certainty` parameter); see `ss.aipe.cv`)

Alternatively, one can specify `Specified.N` to determine the results at a particular sample size (when doing this `Estimated.C.of.V` cannot be specified).

Value

`Data.from.Simulation`	list of the results in matrix form
`Specifications`	specification of the function
`Summary.of.Results`	summary measures of some important descriptive statistics

Note

Returns three lists, where each list has multiple components.

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`)

See Also

`cv`, `ss.aipe.cv`

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(true.variance.trend, error.variance,
variance.true.minus.estimated.trend = NULL, duration, frequency,
width, conf.level = 0.95, trend = "linear", assurance = NULL)
```

Arguments

`true.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 error variance (i.e., σ^2_{ϵ} in Kelley & Rausch, 2011).

`variance.true.minus.estimated.trend`
: The variance of the difference between the mth true change coefficient minus the mth estimated change coefficient (i.e., $\sigma^2_{\hat{\pi}_m - \pi_m}$ from Equation 19 in Kelley & Rausch, 2011).

`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

Returns the necessary sample size for the combination of the desired goals and values of the population parameters for a specific 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 (University of Notre Dame; <KKelley@ND.Edu>)

References

Kelley, K., & Rausch, J. R. (2011). Accuracy in parameter estimation for polynomial change models. *Psychological Methods*.

Examples

```
## Not run:
# An example used in Kelley and Rausch for the expected confidence interval
# width (returns 278). Thus, a necessary sample size of 278 is required when
# the duration of the study will be 4 units and the frequency of measurement
# occasions is 1 year in order for the expected confidence interval
# width to be 0.025 units.

ss.aipe.pcm(true.variance.trend=0.003, error.variance=0.0262, duration=4,
frequency=1, width=0.025, conf.level=.95)

# Now, when incorporating an assurance parameter (returns 316).
# Thus, a necessary sample size of 316 will ensure that the 95% confidence
# interval will be sufficiently narrow (i.e., have a width less than .025 units)
# at least 99% of the time.

ss.aipe.pcm(true.variance.trend=.003, error.variance=.0262, duration=4,
frequency=1, width=.025, conf.level=.95, assurance=.99)

## End(Not run)
```

`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, Random.Regressors, which.width = "Full", p = NULL,
K, degree.of.certainty = NULL, assurance=NULL, certainty=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`)
- `Random.Regressors` an alias for `Random.Predictors`; `Random.Regressors` overrides `Random.Predictors`
- `which.width` defines the width that `width` refers to
- `p` the number of predictor variables
- `K` an alias for `p`; `K` overrides `p`
- `degree.of.certainty` value with which confidence can be placed that describes the likelihood of obtaining a confidence interval less than the value specified (e.e.g, .80, .90, .95)
- `assurance` an alias for `degree.of.certainty`
- `certainty` an alias for `degree.of.certainty`
- `verify.ss` evaluates numerically via an internal Monte Carlo simulation the exact sample size given the specifications
- `Tol` the tolerance of the iterative function `conf.limits.nct` 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 `degree.of.certainty=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 (2007) 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. See the `verify.ss.aipe.R2` function for arguments that can be passed from `ss.aipe.R2` to `verify.ss.aipe.R2`.

Value

`Required.Sample.Size`
: sample size that should be used given the conditions specified.

Note

This function without `verify.SS=FALSE` can be slow to converge when `verify.SS=TRUE`, the function can take some time to converge (e.g., 15 minutes). Most times this will not be the case, but it is possible in some situations.

Author(s)

Ken Kelley (University of Notre Dame; `<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*, 119–136.

Steiger, J. H. & Fouladi, R. T. (1992). R2: A computer program for interval estimation, power calculation, and hypothesis testing for the squared multiple correlation. *Behavior research methods, instruments and computers, 4*, 581–582.

Kelley, K. (2007). Sample size planning for the squared multiple correlation coefficient: Accuracy in parameter estimation via narrow confidence intervals, *manuscripted submitted for publication*.

Kelley, K. & Maxwell, S. E. (2008). Power and accuracy for omnibus and targeted effects: Issues of sample size planning with applications to multiple regression. In P. Alasuuta, J. Brannen, & L. Bickman (Eds.), *Handbook of Social Research Methods* (pp. 166–192). Newbury Park, CA: Sage.

See Also

`ci.R2`, `conf.limits.nct`, `ss.aipe.R2.sensitivity`

Examples

```
## Not run:
# Returned sample size should be considered approximate; exact sample
# size is obtained by specifying the argument 'verify.ss=TRUE' (see below).
# ss.aipe.R2(Population.R2=.50, conf.level=.95, width=.10, which.width="Full",
# p=5, Random.Predictors=TRUE)
# Uncomment to run in order to get exact sample size.
# ss.aipe.R2(Population.R2=.50, conf.level=.95, width=.10, which.width="Full",
# p=5, Random.Predictors=TRUE, verify.ss=TRUE)
```

```
# Same as above, except the predictor variables are considered fixed.
# Returned sample size should be considered approximate; exact sample
# size is obtained by specifying the argument 'verify.ss=TRUE'.
# ss.aipe.R2(Population.R2=.50, conf.level=.95, width=.10, which.width="Full",
# p=5, Random.Predictors=FALSE)
# Uncomment to run in order to get exact sample size.
#ss.aipe.R2(Population.R2=.50, conf.level=.95, width=.10, which.width="Full",
#p=5, Random.Predictors=FALSE, verify.ss=TRUE)


# Returned sample size should be considered approximate; exact sample
# size is obtained by specifying the argument 'verify.ss=TRUE'.
# ss.aipe.R2(Population.R2=.50, conf.level=.95, width=.10, which.width="Full",
# p=5, degree.of.certainty=.85, Random.Predictors=TRUE)
# Uncomment to run in order to get exact sample size.
#ss.aipe.R2(Population.R2=.50, conf.level=.95, width=.10, which.width="Full",
#p=5, degree.of.certainty=.85, Random.Predictors=TRUE, verify.ss=TRUE)


# Same as above, except the predictor variables are considered fixed.
# Returned sample size should be considered approximate; exact sample
# size is obtained by specifying the argument 'verify.ss=TRUE'.
# ss.aipe.R2(Population.R2=.50, conf.level=.95, width=.10, which.width="Full",
# p=5, degree.of.certainty=.85, Random.Predictors=FALSE)
# Uncomment to run in order to get exact sample size.
#ss.aipe.R2(Population.R2=.50, conf.level=.95, width=.10, which.width="Full",
#p=5, degree.of.certainty=.85, Random.Predictors=FALSE, verify.ss=TRUE)

## End(Not run)
```

---

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, Selected.N=NULL,
degree.of.certainty = NULL, assurance=NULL, certainty=NULL,
conf.level = 0.95, Generate.Random.Predictors=TRUE, rho.yx = 0.3,
rho.xx = 0.3,  G = 10000, print.iter = TRUE, ...)
```

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`)
`Selected.N`	selected sample size to use in order to determine distributional properties at a given value of sample size
`degree.of.certainty`	parameter to ensure confidence interval width with a specified degree of certainty
`assurance`	an alias for `degree.of.certainty`
`certainty`	an alias for `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
`...`	for modifying parameters of functions this function calls upon

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 `Selected.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`/indepdent 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

`Results`	a list containing vectors of the empirical results
`Specifications`	outputs the input specifications and required sample size
`Summary`	summary values for the results of the sensitivity analysis (simulation study)

Author(s)

Ken Kelley (University of Notre Dame; <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*, 119–136.

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

Steiger, J. H. & Fouladi, R. T. (1992). R2: A computer program for interval estimation, power calculation, and hypothesis testing for the squared multiple correlation. *Behavior research methods, instruments and computers, 4*, 581–582.

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.

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

See Also

`ci.R2`, `conf.limits.nct`, `ss.aipe.R2`

Examples

```
## Not run:
# Change 'G' to some large number (e.g., G=10,000)
# ss.aipe.R2.sensitivity(True.R2=.5, Estimated.R2=.4, w=.10, p=5, conf.level=0.95,
# G=25)

## End(Not run)
```

`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.k_X.without.k = NULL,
K = NULL, b.k = NULL, width, which.width = "Full", sigma.Y = 1,
sigma.X.k = 1, RHO.XX = NULL, Rho.YX = NULL, which.predictor = NULL,
alpha.lower = NULL, alpha.upper = NULL, conf.level = .95,
degree.of.certainty = NULL, assurance=NULL,  certainty=NULL,
Suppress.Statement = FALSE)
```

Arguments

`Rho2.Y_X` Population value of the squared multiple correlation coefficient

`Rho2.k_X.without.k` Population value of the squared multiple correlation coefficient predicting the *k*th predictor variable from the remaining *K*-1 predictor variables

`K` the number of predictor variables

`b.k` the regression coefficient for the *k*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.k` the population standard deviation of the *k*th *X* variable (i.e., the predictor variable of interest)

`RHO.XX` Population correlation matrix for the *p* predictor variables

`Rho.YX` Population *K* length vector of correlation between the dependent variable (*Y*) and the *K* independent variables

`which.predictor` identifies which of the *K* 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)

`degree.of.certainty`
: degree of certainty that the obtained confidence interval will be sufficiently narrow

`assurance`
: an alias for `degree.of.certainty`

`certainty`
: an alias for `degree.of.certainty`

`Suppress.Statement`
: `TRUE` or `FALSE` statement whether or not a sentence describing the situation defined is printed with the necessary sample size

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 MBESS but has a different naming scheme. See `ss.aipe.reg.coef` for more details.

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>)

References

Kelley, K. & Maxwel, S. E. (2003). Sample size for Multiple Regression: Obtaining regression coefficients that are accuracy, not simply significant. *Psychological Methods, 8*, 305–321.

See Also

`ss.aipe.reg.coef.sensitivity`, `conf.limits.nct`,

`ss.aipe.reg.coef`, `ss.aipe.src`

Examples

```
## Not run:
# 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.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, degree.of.certainty=.85)

## End(Not run)
```

---

ss.aipe.rc.sensitivity   *Sensitivity analysis for sample size planing 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,
degree.of.certainty = NULL, assurance=NULL, certainty=NULL,
G = 1000, print.iter = TRUE)
```

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 using `Estimated.Rho.YX` and `Estimated.RHO.XX`)
- `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)
degree.of.certainty	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).
assurance	an alias for degree.of.certainty
certainty	an alias for degree.of.certainty
G	the number of generations/replication of the simulation student 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

Details

Direct specification of True.Rho.YX and True.RHO.XX is necessary, even if one is interested in a single regression coefficient, so that the covariance/correlation structure can be specified when when the simulation student within the function runs.

Value

Results	a matrix containing the empirical results from each of the G replication of the simulation
Specifications	a list of the input specifications and the required sample size
Summary.of.Results	summary values for the results of the sensitivity analysis (simulation study) given the input specification

Note

Note that when True.Rho.YX=Estimated.Rho.YX and True.RHO.XX=Estimated.RHO.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 MBESS for more details.

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>)

References

Kelley, K. & Maxwell, S. E. (2003). Sample size for Multiple Regression: Obtaining regression coefficients that are accuracy, not simply significant. *Psychological Methods, 8*, 305–321.

See Also

`ss.aipe.reg.coef.sensitivity`, `ss.aipe.src.sensitivity`,

`ss.aipe.reg.coef`, `ci.reg.coef`

`ss.aipe.reg.coef` *Sample size necessary for the accuracy in parameter estimation approach for a regression coefficient of interest*

Description

A function used to plan sample size from the accuracy in parameter estimation approach for a regression coefficient of interest given the input specification.

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=.95, degree.of.certainty=NULL, assurance=NULL,
certainty=NULL, Suppress.Statement=FALSE)
```

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`	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
`Noncentral`	specify with a `TRUE` or `FALSE` statement whether or not the noncentral approach to sample size planning should be used

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)

degree.of.certainty
degree of certainty that the obtained confidence interval will be sufficiently narrow

assurance an alias for degree.of.certainty

certainty an alias for degree.of.certainty

Suppress.Statement
TRUE/FALSE statement whether or not a sentence describing the situation defined is printed with the necessary sample size

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.

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>)

References

Kelley, K. & Maxwel, S. E. (2003). Sample size for Multiple Regression: Obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods, 8*, 305–321.

See Also

ss.aipe.reg.coef.sensitivity, conf.limits.nct

Examples

```
## Not run:
# 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.reg.coef(width=.1, which.width="Full", sigma.Y=1, sigma.X=1, RHO.XX=RHO.XX,
# Rho.YX=Rho.YX, which.predictor=1, Noncentral=FALSE, conf.level=1-.05,
# degree.of.certainty=NULL, Suppress.Statement=FALSE)

# ss.aipe.reg.coef(width=.1, which.width="Full", sigma.Y=1, sigma.X=1, RHO.XX=RHO.XX,
```

```
# Rho.YX=Rho.YX, which.predictor=1, Noncentral=FALSE, conf.level=1-.05,
# degree.of.certainty=.85, Suppress.Statement=FALSE)

# ss.aipe.reg.coef(width=.1, which.width="Full", sigma.Y=1, sigma.X=1, RHO.XX=RHO.XX,
# Rho.YX=Rho.YX, which.predictor=1, Noncentral=TRUE, conf.level=1-.05,
# degree.of.certainty=NULL, Suppress.Statement=FALSE)

# ss.aipe.reg.coef(width=.1, which.width="Full", sigma.Y=1, sigma.X=1, RHO.XX=RHO.XX,
# Rho.YX=Rho.YX, which.predictor=1, Noncentral=TRUE, conf.level=1-.05,
# degree.of.certainty=.85, Suppress.Statement=FALSE)
## End(Not run)
```

`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

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,
degree.of.certainty = NULL, assurance=NULL, certainty=NULL,
 G = 1000, print.iter = TRUE)
```

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 using `Estimated.Rho.YX` and `Estimated.RHO.XX`)

`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)

`degree.of.certainty`
: degree of certainty that the obtained confidence interval will be sufficiently narrow

`assurance`
: an alias for `degree.of.certainty`

`certainty`
: an alias for `degree.of.certainty`

`G`
: the number of generations/replication of the simulation student within the function

`print.iter`
: specify with a `TRUE/FALSE` statement if the iteration number should be printed as the simulation within the function runts

Details

Direct specification of `True.Rho.YX` and `True.RHO.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 student within the function runs.

Value

`Results`
: a matrix containing the empirical results from each of the `G` replications of the simulation

`Specifications`
: a list of the input specifications and the required sample size

`Summary.of.Results`
: summary values for the results of the sensitivity analysis (simulation study) given the input specification

Note

Note that when `True.Rho.YX=Estimated.Rho.YX` and `True.RHO.XX=Estimated.RHO.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 (University of Notre Dame; <KKelley@ND.Edu>)

References

Kelley, K. & Maxwell, S. E. (2003). Sample size for Multiple Regression: Obtaining regression coefficients that are accuracy, not simply significant. *Psychological Methods, 8*, 305–321.

See Also

ss.aipe.reg.coef, ci.reg.coef

---

ss.aipe.reliability — *Sample Size Planning for Accuracy in Parameter Estimation for Reliability Coefficients.*

Description

This function determines a necessary sample size so that the expected confidence interval width for the alpha coefficient or omega coefficient is sufficiently narrow (when assurance=NULL) or 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; assurance=.85). This function calculates coefficient alpha based on McDonald's (1999) formula for coefficient alpha, also known as Guttman-Cronbach alpha. It also uses coefficient omega from McDonald (1999). When the 'Parallel' or 'True Score' model is used, coefficient alpha is calculated. When the 'Congeneric' model is used, coefficient omega is calculated.

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 type of measurement model (e.g., "parallel items", "true-score equivalent", or "congeneric model") for a homogeneous single common factor test
type	the type of method to base the formation of the confidence interval on, either the "Factor Analytic" (McDonald, 1999) or "Normal Theory" (van Zyl, Neudecker, & Nel, 2000)
width	the desired full width of the confidence interval
S	a symmetric covariance matrix
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 specificied degree of certainty
data	the data set that the reliability coefficient is obtained from
i	number of items

`cor.est`	the estimated inter-item correlation
`lambda`	the vector of population factor loadings
`psi.square`	the vector of population error variances
`initial.iter`	the number of initial iterations or generations/replications of the simulation study within the function
`final.iter`	the number of final iterations or generations/replications of the simulation study
`start.ss`	the initial sample size to start the simulation at
`verbose`	shows extra information one the current sample size and current level of assurance; helpful if the function gets stuck in a long iterative process

Details

Use `verbose=TRUE` if the function is taking a very long time to provide an answer.

Value

`Required.Sample.Size`	the necessary sample size
`width`	the specified full width of the confidence interval
`specified.assurance`	the specified degree of certainty
`empirical.assurance`	the empirical assurance based on the necessary sample size returned
`final.iter`	the specified number of iterations in the simulation study

Warning

In some conditions, you may receive a warning, such as "`In sem.default(ram = ram, S = S, N = N, param.names = par` This indicates that the model likely did not converge. In certain conditions this may occur because the model is not being fit well due to 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)

Leann J. Terry (Indiana University; <ljterry@Indiana.Edu>); Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>)

References

McDonald, R. P. (1999). *Test theory: A unified approach.* Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

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.

See Also

CFA.1; sem; ci.reliability;

Examples

```
## Not run:
ss.aipe.reliability (model='Parallel', type='Normal Theory', width=.1, i=6,
                cor.est=.3, psi.square=.2, conf.level=.95, assurance=NULL, initial.iter=500,
                    final.iter=5000)

# Same as above but now 'assurance' is used.
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=500,
final.iter=5000)


# Similar to the above but now the "True Score" model is used. Note how the psi.square changes
# from a scalar to a vector of length i (number of items).
# Also note, however, that cor.est is a single value (due to the true-score model specified)
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=500, final.iter=5000)

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=500, final.iter=5000)

# Now, a congeneric model is used with the factor analytic appraoch. This is likely the
# most realistic scenario (and maps onto the ideas of Coefficient Omega).
ss.aipe.reliability (model='Congeneric', type='Factor Analytic', width=.1, i=5,
lambda=c(.4, .4, .3, .3, .5), psi.square=c(.2, .4, .3, .3, .2), conf.level=.95,
assurance=.85, initial.iter=1000, final.iter=5000)

# Now, the presumed population matrix among the items is used.
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=1000, final.iter=5000)
```

```
## End(Not run)
```

---

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 (University of Notre Dame; <KKelley@ND.Edu>) and Keke Lai

See Also

ci.rmsea

Examples

```
## Not run:
# ss.aipe.rmsea(RMSEA=.035, df=50, width=.05, conf.level=.95)

## End(Not run)
```

`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 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, save.file="sim.results.txt", ...)
```

Arguments

`width`	desired confidence interval width for the model parameter of interest
`model`	the model the researcher proposes, may or may not be the true model. This argument should be an RAM (reticular action model; e.g., McArdle & McDonald, 1984) specification of a structural equation model, and should be of class `mod`. The model is specified in the same manner as does the `sem` package; see `sem` and `specify.model` for detailed documentation about model specifications in the RAM notation.
`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 `Sigma` should be the same as the manifest variables in `model`.
`N`	if `N` is specified, random sample of the specified `N` size will be generated. Otherwise the sample size is calculated with the sample size planning method with the goal that the expected width of a confidence interval for population RMSEA is no larger than desired.
`conf.level`	confidence level (i.e., 1- Type I error rate)
`G`	number of replications in the Monte Carlo simulation
`save.file`	the name of the file that simulation results will be saved to
`...`	allows one to potentially include parameter values for inner functions

Details

This function implements the sample size planning methods proposed in Kelley and Lai (2010). It depends on the function `sem` in the `sem` package to fit the proposed model to random data, and uses the same notation to specify SEM models as does `sem`. Please refer to `sem` for more detailed documentation about model specifications, the RAM notation, and model fitting techniques. For technical discussion on how to obtain the model implied covariance matrix in the RAM notation given model parameters, see McArdle and McDonald (1984)

Value

`successful.replication`
: the number of successful replications

`w`
: the `G` random confidence interval widths

`RMSEA.hat`
: the `G` estimated RMSEA values based on the `G` random samples

`sample.size`
: the sample size calculated

`df`
: degrees of freedom of the proposed model

`RMSEA.pop`
: the input RMSEA value that is used to calculated the necessary sample size

`desired.width`
: desired confidence interval width

`mean.width`
: mean of the random confidence interval widths

`median.width`
: median of the random confidence interval widths

`assurance`
: the proportion of confidence interval widths narrower than desired

`quantile.width`
: 99, 97, 95, 90, 80, 70, and 60 percentiles of the random confidence interval widths

`alpha.upper`
: the upper empirical Type I error rate

`alpha.lower`
: the lower empirical Type I error rate

`alpha`
: total empirical Type I error rate

`conf.level`
: confidence level

`sim.results.txt`
: a text file that saves the simulation results; it updates after each replication. 'sim.results.txt' is the default file name

Note

Sometimes this function jumps out of the loop before it finishes the simulation. The reason is because the `sem` function that this function calls to fit the model fails to converge when searching for maximum likelihood estimates of model parameters. Since the results in previous replications are saved, the user can start this function again, and specify the number of replications (i.e., `G`) to be the desired total number of replications minus the number of previous successful replications.

Author(s)

Keke Lai (University of California – Merced) and Ken Kelley (University of Notre Dame; <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.

Fox, J. (2006). Structural equation modeling with the sem package in R. *Structural Equation Modeling, 13*, 465–486.

Kelley, K., & Lai, K. (2010). Accuracy in parameter estimation for the root mean square of approximation: Sample size planning for narrow confidence intervals. *Manuscript under review*.

McArdle, J. J., & McDonald, R. P. (1984). Some algebraic properties of the reticular action model. *British Journal of Mathematical and Statistical Psychology, 37*, 234–251.

See Also

sem; specify.model; ss.aipe.rmsea; theta.2.Sigma.theta; Sigma.2.SigmaStar

Examples

```
## Not run:
#########################
EXAMPLE 1
#########################
# To replicate the simulation in the first panel, second column of
# Table 2 (i.e., population RMSEA=0.0268, df=23, desired width=0.02)
# in Lai and Kelley (2010), the following steps can be used.

## STEP 1: Obtain the (correct) population covariance matrix implied by Model 2
# This requires the model and its population model parameter values.
library(MASS)
library(sem)

# Specify Model 2 in the RAM notation
model.2<-specifyModel()
xi1 -> y1, lambda1, 1
xi1 -> y2, NA, 1
xi1 -> y3, lambda2, 1
xi1 -> y4, lambda3, 0.3
eta1 -> y4, lambda4, 1
eta1 -> y5, NA, 1
eta1 -> y6, lambda5, 1
eta1 -> y7, lambda6, 0.3
eta2 -> y6, lambda7, 0.3
eta2 -> y7, lambda8, 1
eta2 -> y8, NA, 1
eta2 -> y9, lambda9, 1
xi1 -> eta1, gamma11, 0.6
eta1 -> eta2, beta21, 0.6
xi1 <-> xi1, phi11, 0.49
eta1 <-> eta1, psi11, 0.3136
eta2 <-> eta2, psi22, 0.3136
y1 <-> y1, delta1, 0.51
y2 <-> y2, delta2, 0.51
y3 <-> y3, delta3, 0.51
y4 <-> y4, delta4, 0.2895
y5 <-> y5, delta5, 0.51
y6 <-> y6, delta6, 0.2895
y7 <-> y7, delta7, 0.2895
y8 <-> y8, delta8, 0.51
y9 <-> y9, delta9, 0.51


# To inspect the specified model
model.2

# Specify model parameter values
```

```
theta <- c(1, 1, 0.3, 1,1, 0.3, 0.3, 1, 1, 0.6, 0.6,
0.49, 0.3136, 0.3136, 0.51, 0.51, 0.51, 0.2895, 0.51, 0.2895, 0.2895, 0.51, 0.51)

names(theta) <- c("lambda1","lambda2","lambda3",
"lambda4","lambda5","lambda6","lambda7","lambda8","lambda9",
"gamma11", "beta21",
"phi11", "psi11", "psi22",
"delta1","delta2","delta3","delta4","delta5","delta6","delta7",
"delta8","delta9")

res<-theta.2.Sigma.theta(model=model.2, theta=theta,
latent.vars=c("xi1", "eta1","eta2"))

Sigma.theta <- res$Sigma.theta
# Then 'Sigma.theta' is the (true) population covariance matrix

## STEP 2: Create a misspecified model
# The following model is misspecified in the same way as did Lai and Kelley (2010)
# with the goal to obtain a relatively small population RMSEA

model.2.mis<-specifyModel()
xi1 -> y1, lambda1, 1
xi1 -> y2, NA, 1
xi1 -> y3, lambda2, 1
xi1 -> y4, lambda3, 0.3
eta1 -> y4, lambda4, 1
eta1 -> y5, NA, 1
eta1 -> y6, lambda5, 0.96
eta2 -> y6, lambda7, 0.33
eta2 -> y7, lambda8, 1.33
eta2 -> y8, NA, 1
eta2 -> y9, lambda9, 1
xi1 -> eta1, gamma11, 0.6
eta1 -> eta2, beta21, 0.65
xi1 <-> xi1, phi11, 0.49
eta1 <-> eta1, psi11, 0.3136
eta2 <-> eta2, psi22, 0.23
y1 <-> y1, delta1, 0.51
y2 <-> y2, delta2, 0.51
y3 <-> y3, delta3, 0.51
y4 <-> y4, delta4, 0.2895
y5 <-> y5, delta5, 0.51
y6 <-> y6, delta6, 0.29
y7 <-> y7, delta7, 0.22
y8 <-> y8, delta8, 0.56
y9 <-> y9, delta9, 0.56


# To verify the population RMSEA of this misspecified model
fit<-sem(ram=model.2.mis, S=Sigma.theta, N=1000000)
summary(fit)$RMSEA

## STEP 3: Conduct the simulation
```

```
# The number of replications is set to a very small value just to demonstrate
# and save time. Real simulation studies require a larger number (e.g., 500, 1,000)

ss.aipe.rmsea.sensitivity(width=0.02, model=model.2.mis, Sigma=Sigma.theta, G=10)

## STEP 3+: In cases where this function stops before it finishes the simulation
# Suppose it stops at the 7th replication. The text
# file "results_ss.aipe.rmsea.sensitivity.txt" saves the results in all
# previous replications; in this case it contains 6 replications since
# the simulation stopped at the 7th. The user can start this function again and specify
# 'G' to 4 (i.e., 10-6). New results will be appended to previous ones in the same file.

ss.aipe.rmsea.sensitivity(width=0.02, model=model.2.mis, Sigma=Sigma.theta, G=4)

########################################
EXAMPLE 2
########################################
# In addition to create a misspecified model by changing the model
# parameters in the true model as does Example 1, a misspecified
# model can also be created with the Cudeck-Browne (1992) procedure.
# This procedure is implemented in the 'Sigma.2.SigmaStar( )' function in
# the MBESS package. Please refer to the help file of 'Sigma.2.SigmaStar( )'
# for detailed documentation.

## STEP 1: Specify the model
# This model is the same as the model in the first step of Example 1, but the
# model-implied population covariance matrix is no longer the true population
# covariance matrix. The true population covariance matrix will be generated
# in Step 2 with the Cudeck-Browne procedure.
library(MASS)
library(sem)

model.2<-specifyModel()
xi1 -> y1, lambda1, 1
xi1 -> y2, NA, 1
xi1 -> y3, lambda2, 1
xi1 -> y4, lambda3, 0.3
eta1 -> y4, lambda4, 1
eta1 -> y5, NA, 1
eta1 -> y6, lambda5, 1
eta1 -> y7, lambda6, 0.3
eta2 -> y6, lambda7, 0.3
eta2 -> y7, lambda8, 1
eta2 -> y8, NA, 1
eta2 -> y9, lambda9, 1
xi1 -> eta1, gamma11, 0.6
eta1 -> eta2, beta21, 0.6
xi1 <-> xi1, phi11, 0.49
eta1 <-> eta1, psi11, 0.3136
eta2 <-> eta2, psi22, 0.3136
y1 <-> y1, delta1, 0.51
y2 <-> y2, delta2, 0.51
y3 <-> y3, delta3, 0.51
```

```
y4 <-> y4, delta4, 0.2895
y5 <-> y5, delta5, 0.51
y6 <-> y6, delta6, 0.2895
y7 <-> y7, delta7, 0.2895
y8 <-> y8, delta8, 0.51
y9 <-> y9, delta9, 0.51



theta <- c(1, 1, 0.3, 1,1, 0.3, 0.3, 1, 1, 0.6, 0.6,
0.49, 0.3136, 0.3136, 0.51, 0.51, 0.51, 0.2895, 0.51, 0.2895, 0.2895, 0.51, 0.51)

names(theta) <- c("lambda1","lambda2","lambda3",
"lambda4","lambda5","lambda6","lambda7","lambda8","lambda9",
"gamma11", "beta21",
"phi11", "psi11", "psi22",
"delta1","delta2","delta3","delta4","delta5","delta6","delta7",
"delta8","delta9")

## STEP 2: Create the true population covariance matrix, so that (a) the model fits
# to this covariance matrix with specified discrepancy; (b) the population model
# parameters (the object 'theta') is the minimizer in fitting the model to the true
# population covariance matrix.

# Since the desired RMSEA is 0.0268 and the df is 22, the MLE discrepancy value
# is specified to be 22*0.0268*0.0268, given the definition of RMSEA.

res <- Sigma.2.SigmaStar(model=model.2, model.par=theta,
latent.var=c("xi1", "eta1", "eta2"), discrep=22*0.0268*0.0268)

Sigma.theta.star <- res$Sigma.star

# To verify that the population RMSEA is 0.0268
res2 <- sem(ram=model.2, S=Sigma.theta.star, N=1000000)
summary(res2)$RMSEA

## STEP 3: Conduct the simulation
# Note although Examples 1 and 2 have the same population RMSEA, the
# model df and true population covariance matrix are different. Example 1
# uses 'model.2.mis' and 'Sigma.theta', whereas Example 2 uses 'model.2'
# and 'Sigma.theta.star'. Since the df is different, it requires a different sample
# size to achieve the same desired confidence interval width.
ss.aipe.rmsea.sensitivity(width=0.02, model=model.2, Sigma=Sigma.theta.star, G=10)

## End(Not run)
```

`ss.aipe.sc` *Sample size planning for Accuracy in Parameter Estimation (AIPE) of the standardized contrast in ANOVA*

Description

A function to calculate the appropriate sample size per group for the standardized contrast in ANOVA such that the width of the confidence interval is sufficiently narrow.

Usage

```
ss.aipe.sc(psi, c.weights, width, conf.level = 0.95,
assurance = NULL, certainty = NULL, ...)
```

Arguments

`psi`	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)
`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)
`certainty`	an alias for `assurance`
`...`	allows one to potentially include parameter values for inner functions

Value

`n`	necessary sample size *per group*

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`); Keke Lai

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*, 532–574.

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 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.

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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.

Lai, K., & Kelley, K. (2007). Sample size planning for standardized ANCOVA and ANOVA contrasts: Obtaining narrow confidence intervals. *Manuscript submitted for publication*.

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`, `conf.limits.nct`, `ss.aipe.c`

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 gorup 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=.6, c.weights=c(.5, .5, -.5, -.5, 0), width=.4, assurance=.90)
```

`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 = NULL, ratio = NULL, rho = NULL, divisor = "s.ancova",
c.weights, width, conf.level = 0.95, 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`	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)
`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 arguments about the effect sizes need to be specified. If `divisor="s.ancova"` is used in the argument, then input either (a) `psi`, or (b) `Psi` and `s.ancova`. If `divisor="s.anova"` is used in the argument, possible specifications are (a) `Psi`, `s.ancova`, and `s.anova`; (b) `psi`, and `ratio`; (c) `psi`, and `rho`.

Value

This function returns the sample size *per group*.

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 Montre Carlo simulations with function `ss.aipe.sc.ancova.sensitivity`, as discussed in Lai & Kelley (under review).

Author(s)

Keke Lai (University of California–Merced)

References

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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.

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.

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`

Examples

```
## Not run:
ss.aipe.sc.ancova(psi=.8, width=.5, c.weights=c(.5, .5, 0, -1))

ss.aipe.sc.ancova(psi=.8, ratio=.6, width=.5,
c.weights=c(.5, .5, 0, -1), divisor="s.anova")

ss.aipe.sc.ancova(psi=.5, rho=.4, width=.3,
c.weights=c(.5, .5, 0, -1), divisor="s.anova")

## End(Not run)
```

---

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, selected.n = NULL, mu.x = 0,
sigma.x = 1, rho, divisor = "s.ancova", assurance = NULL,
conf.level = 0.95, G = 10000, print.iter = TRUE, detail = TRUE, ...)
```

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
`selected.n`	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
- `detail` whether the user needs a detailed (`TRUE`) or brief (`FALSE`) report of the simulation results; the detail report includes all the raw data in the simulations
- `...` 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

- `psi.obs` observed standardized contrast in each iteration
- `Full.Width` vector of the full confidence interval width
- `Width.from.psi.obs.Lower` vector of the lower confidence interval width
- `Width.from.psi.obs.Upper` vector of the upper confidence interval width
- `Type.I.Error.Upper` iterations where a Type I error occurred on the upper end of the confidence interval
- `Type.I.Error.Lower` iterations where a Type I error occurred on the lower end of the confidence interval
- `Type.I.Error` iterations where a Type I error happens
- `Lower.Limit` the lower limit of the obtained confidence interval
- `Upper.Limit` the upper limit of the obtained confidence interval
- `replications` number of replications of the simulation

`True.psi` population standardized contrast

`Estimated.psi` estimated standardized contrast

`Desired.Width` the desired full width of the obtained confidence interval

`assurance` the value assigned to the argument `assurance`

`Sample.Size.per.Group`
sample size *per group*

`Number.of.Groups`
number of groups

`mean.full.width`
mean width of the obtained full confidence intervals

`median.full.width`
median width of the obtained full confidence intervals

`sd.full.width` standard deviation of the widths of the obtained full confidence intervals

`Pct.Width.obs.NARROWER.than.desired`
percentage of the obtained full confidence interval widths that are narrower than the desired width

`mean.Width.from.psi.obs.Lower`
mean lower width of the obtained confidence intervals

`mean.Width.from.psi.obs.Upper`
mean upper width of the obtained confidence intervals

`Type.I.Error.Upper`
Type I error rate from the upper side

`Type.I.Error.Lower`
Type I error rate from the lower side

`Type.I.Error` Type I error rate

Author(s)

Keke Lai

References

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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.

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.

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` `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, selected.n = NULL, assurance = NULL, certainty=NULL,
conf.level = 0.95, G = 10000, print.iter = TRUE, detail = TRUE, ...)
```

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
`selected.n`	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)
`certainty`	an alias for `assurance`
`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
`detail`	whether the user needs a detailed (`TRUE`) or brief (`FALSE`) report of the simulation results; the detailed report includes all the raw data in the simulations
`...`	allows one to potentially include parameter values for inner functions

Value

`psi.obs`	observed standardized contrast in each iteration
`Full.Width`	vector of the full confidence interval width
`Width.from.psi.obs.Lower`	vector of the lower confidence interval width

`Width.from.psi.obs.Upper`
: vector of the upper confidence interval width

`Type.I.Error.Upper`
: iterations where a Type I error occurred on the upper end of the confidence interval

`Type.I.Error.Lower`
: iterations where a Type I error occurred on the lower end of the confidence interval

`Type.I.Error`
: iterations where a Type I error happens

`Lower.Limit`
: the lower limit of the obtained confidence interval

`Upper.Limit`
: the upper limit of the obtained confidence interval

`replications`
: number of replications of the simulation

`True.psi`
: population standardized contrast

`Estimated.psi`
: estimated standardized contrast

`Desired.Width`
: the desired full width of the obtained confidence interval

`assurance`
: the value assigned to the argument `assurance`

`Sample.Size.per.Group`
: sample size per group

`Number.of.Groups`
: number of groups

`mean.full.width`
: mean width of the obtained full conficence intervals

`median.full.width`
: median width of the obtained full confidence intervals

`sd.full.width`
: standard deviation of the widths of the obtained full confidence intervals

`Pct.Width.obs.NARROWER.than.desired`
: percentage of the obtained full confidence interval widths that are narrower than the desired width

`mean.Width.from.psi.obs.Lower`
: mean lower width of the obtained confidence intervals

`mean.Width.from.psi.obs.Upper`
: mean upper width of the obtained confidence intervals

`Type.I.Error.Upper`
: Type I error rate from the upper side

`Type.I.Error.Lower`
: Type I error rate from the lower side

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`); Keke Lai (University of California – Merced)

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*, 532–574.

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 2*, 107–128.

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

Kelley, K., & Rausch, J. R. (2006). Sample size planning for the standardized mean difference: Accuracy in Parameter Estimation via narrow confidence intervals. P*sychological Methods, 11* (4), 363–385.

Lai, K., & Kelley, K. (2007). Sample size planning for standardized ANCOVA and ANOVA contrasts: Obtaining narrow confidence intervals. *Manuscript submitted for publication*.

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 where no significance tests?* (pp. 221–257). Mahwah, NJ: Lawrence Erlbaum.

See Also

`ss.aipe.sc`, `ss.aipe.c`, `conf.limits.nct`

`ss.aipe.sem.path` *Sample size planning for SEM targeted effects*

Description

Plan sample size for structural equation models so that the confidence intervals for the model parameters of interest are sufficiently narrow

Usage

```
ss.aipe.sem.path(model, Sigma, desired.width, which.path,
conf.level = 0.95, assurance = NULL, ...)
```

Arguments

`model`	an RAM (reticular action model; e.g., McArdle & McDonald, 1984) specification of a structural equation model, and should be of class `mod`. The model is specified in the same manner as the `sem` package; see `sem` and `specify.model` for detailed documentation about model specifications in the RAM notation.
`Sigma`	estimated population covariance matrix of the manifest variables
`desired.width`	desired confidence interval width for the model parameter of interest
`which.path`	the name of the model parameter of interest, presented in double quotation marks

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)
...	allows one to potentially include parameter values for inner functions

Details

This function implements the sample size planning methods proposed in Lai and Kelley (2010). It depends on the function sem in the sem package to calculate the expected information matrix, and uses the same notation to specify SEM models as does sem. Please refer to sem for more detailed documentations about model specification, the RAM notation, and model fitting techniques. For technical discussion on how to obtain the model implied covariance matrix in the RAM notation given model parameters, see McArdle and McDonald (1984).

Value

parameters	the names of the model parameters
path.index	the index of the model parameter of interest
sample.size	the necessary sample size calculated
obs.vars	the names of the observed variables
var.theta.j	the population variance of the model parameter of interest at the calculated sample size

Author(s)

Keke Lai (University of California–Merced)

References

Fox, J. (2006). Structural equation modeling with the sem package in R. *Structural Equation Modeling, 13*, 465–486.

Lai, K., & Kelley, K. (in press). Accuracy in parameter estimation for targeted effects in structural equation modeling: Sample size planning for narrow confidence intervals. *Psychological Methods*.

McArdle, J. J., & McDonald, R. P. (1984). Some algebraic properties of the reticular action model. *British Journal of Mathematical and Statistical Psychology, 37*, 234–251.

See Also

sem; specify.model; theta.2.Sigma.theta; ss.aipe.sem.path.sensitiv

Examples

```
## Not run:
# Suppose the model of interest is Model 2 in the simulation study
# in Lai and Kelley (2010), and the goal is to obtain a 95% confidence
# interval for 'beta21' no wider than 0.3. The necessary sample size
# can be calculated as follows.
```

```
library(sem)

# specify a model object in the RAM notation
model.2<-specifyModel()
xi1 -> y1, lambda1, 1
xi1 -> y2, NA, 1
xi1 -> y3, lambda2, 1
xi1 -> y4, lambda3, 0.3
eta1 -> y4, lambda4, 1
eta1 -> y5, NA, 1
eta1 -> y6, lambda5, 1
eta1 -> y7, lambda6, 0.3
eta2 -> y6, lambda7, 0.3
eta2 -> y7, lambda8, 1
eta2 -> y8, NA, 1
eta2 -> y9, lambda9, 1
xi1 -> eta1, gamma11, 0.6
eta1 -> eta2, beta21, 0.6
xi1 <-> xi1, phi11, 0.49
eta1 <-> eta1, psi11, 0.3136
eta2 <-> eta2, psi22, 0.3136
y1 <-> y1, delta1, 0.51
y2 <-> y2, delta2, 0.51
y3 <-> y3, delta3, 0.51
y4 <-> y4, delta4, 0.2895
y5 <-> y5, delta5, 0.51
y6 <-> y6, delta6, 0.2895
y7 <-> y7, delta7, 0.2895
y8 <-> y8, delta8, 0.51
y9 <-> y9, delta9, 0.51


# to inspect the specified model
model.2

# one way to specify the population covariance matrix is to first
# specify path coefficients and then calcualte the model-implied
# covariance matrix
theta <- c(1, 1, 0.3, 1,1, 0.3, 0.3, 1, 1, 0.6, 0.6,
0.49, 0.3136, 0.3136, 0.51, 0.51, 0.51, 0.2895, 0.51, 0.2895, 0.2895, 0.51, 0.51)

names(theta) <- c("lambda1","lambda2","lambda3",
"lambda4","lambda5","lambda6","lambda7","lambda8","lambda9",
"gamma11", "beta21",
"phi11", "psi11", "psi22",
"delta1","delta2","delta3","delta4","delta5","delta6","delta7",
"delta8","delta9")

res<-theta.2.Sigma.theta(model=model.2, theta=theta,
latent.vars=c("xi1", "eta1","eta2"))

Sigma.theta <- res$Sigma.theta
# thus 'Sigma.theta' is the input covariance matrix for sample size
```

```
# planning procedure.

# the necessary sample size can be calculated as follows.
# ss.aipe.sem.path(model=model.2, Sigma=Sigma.theta,
# desired.width=0.3, which.path="beta21")

## End(Not run)
```

---

ss.aipe.sem.path.sensitiv *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.sensitiv(model, est.Sigma, true.Sigma = est.Sigma,
which.path, desired.width, N=NULL, conf.level = 0.95, assurance = NULL,
G = 100, ...)
```

Arguments

`model`	the model the researcher proposes, may or may not be the true model. This argument should be an RAM (reticular action model; e.g., McArdle & McDonald, 1984) specification of a structural equation model, and should be of class `mod`. The model is specified in the same manner as does the `sem` package; see `sem` and `specify.model` for detailed documentation about model specifications in the RAM notation.
`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 manifest variables in `est.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 `est.Sigma` should be the same as the manifest variables in `est.model`.
`which.path`	the name of the model parameter of interest, and must be in a double quote
`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
`...`	allows one to potentially include parameter values for inner functions

Details

This function implements the sample size planning methods proposed in Lai and Kelley (2010). It depends on the function `sem` in the `sem` package to calculate the expected information matrix, and uses the same notation to specify SEM models as does `sem`. Please refer to `sem` for more detailed documentation about model specifications, the RAM notation, and model fitting techniques. For technical discussion on how to obtain the model implied covariance matrix in the RAM notation given model parameters, see McArdle and McDonald (1984).

Value

`w`	the `G` random confidence interval widths
`sample.size`	the sample size calculated
`path.of.interest`	name of the model parameter of interest
`desired.width`	desired confidence interval width
`mean.width`	mean of the `G` random confidence interval widths
`median.width`	median of the `G` random confidence interval widths
`quantile.width`	99, 95, 90, 85, 80, 75, 70, and 60 percentiles of the `G` random confidence interval widths
`width.less.than.desired`	the proportion of confidence interval widths narrower than desired
`Type.I.err.upper`	the upper empirical Type I error rate
`Type.I.err.lower`	the lower empirical Type I error rate
`Type.I.err`	total empirical Type I error rate
`conf.level`	confidence level
`rep`	successful replications

Note

Sometimes the simulation stops in the middle of fitting the model to the random data. The reason is that `nlm`, the function `sem` calls to fit the model, fails to converge. We suggest using the `try` function in simulation so that the simulation can proceed with unsuccessful iterations.

Author(s)

Keke Lai (University of California – Merced) and Ken Kelley <kkelley@nd.edu>

References

Fox, J. (2006). Structural equation modeling with the sem package in R. *Structural Equation Modeling, 13*, 465–486.

Lai, K., & Kelley, K. (in press). Accuracy in parameter estimation for targeted effects in structural equation modeling: Sample size planning for narrow confidence intervals. *Psychological Methods*.

McArdle, J. J., & McDonald, R. P. (1984). Some algebraic properties of the reticular action model. *British Journal of Mathematical and Statistical Psychology, 37*, 234–251.

See Also

sem; specify.model; theta.2.Sigma.theta; ss.aipe.sem.path

Examples

```
## Not run:
# Suppose the model of interest is Model 2 of the simulation study in
# Lai and Kelley (2010), and the goal is to obtain a 95% confidence
# interval for 'beta21' no wider than 0.3.

library(sem)

# specify a model object in the RAM notation
model.2<-specifyModel()
xi1 -> y1, lambda1, 1
xi1 -> y2, NA, 1
xi1 -> y3, lambda2, 1
xi1 -> y4, lambda3, 0.3
eta1 -> y4, lambda4, 1
eta1 -> y5, NA, 1
eta1 -> y6, lambda5, 1
eta1 -> y7, lambda6, 0.3
eta2 -> y6, lambda7, 0.3
eta2 -> y7, lambda8, 1
eta2 -> y8, NA, 1
eta2 -> y9, lambda9, 1
xi1 -> eta1, gamma11, 0.6
eta1 -> eta2, beta21, 0.6
xi1 <-> xi1, phi11, 0.49
eta1 <-> eta1, psi11, 0.3136
eta2 <-> eta2, psi22, 0.3136
y1 <-> y1, delta1, 0.51
y2 <-> y2, delta2, 0.51
y3 <-> y3, delta3, 0.51
y4 <-> y4, delta4, 0.2895
y5 <-> y5, delta5, 0.51
y6 <-> y6, delta6, 0.2895
y7 <-> y7, delta7, 0.2895
y8 <-> y8, delta8, 0.51
y9 <-> y9, delta9, 0.51
```

```
# to inspect the specified model
model.2

# one way to specify the population covariance matrix is to
# first specify path coefficients and then calcualte the
# model-implied covariance matrix
theta <- c(1, 1, 0.3, 1,1, 0.3, 0.3, 1, 1, 0.6, 0.6,
0.49, 0.3136, 0.3136, 0.51, 0.51, 0.51, 0.2895, 0.51, 0.2895, 0.2895, 0.51, 0.51)

names(theta) <- c("lambda1","lambda2","lambda3",
"lambda4","lambda5","lambda6","lambda7","lambda8","lambda9",
"gamma11", "beta21",
"phi11", "psi11", "psi22",
"delta1","delta2","delta3","delta4","delta5","delta6","delta7",
"delta8","delta9")

res<-theta.2.Sigma.theta(model=model.2, theta=theta,
latent.vars=c("xi1", "eta1","eta2"))

Sigma.theta <- res$Sigma.theta
# thus 'Sigma.theta' is the input covariance matrix for sample size planning procedure.

# the necessary sample size can be calculated as follows.
# ss.aipe.sem.path(model=model.2, Sigma=Sigma.theta,
# desired.width=0.3, which.path="beta21")

# to verify the sample size calculated
# ss.aipe.sem.path.sensitiv(est.model=model.2, est.Sigma=Sigma.theta,
# which.path="beta21", desired.width=0.3, G = 300)

# suppose the true covariance matrix ('var(X)' below) is in fact
# a point close to 'Sigma.theta':

# X<-mvrnorm(n=1000, mu=rep(0,9), Sigma=Sigma.pop)
# var(X)
# ss.aipe.sem.path.sensitiv(est.model=model.2, est.Sigma=Sigma.theta,
# true.Sigma=var(X), which.path="beta21", desired.width=0.3, G=300)

## End(Not run)
```

`ss.aipe.sm` *Sample size planning for Accuracy in Parameter Estimation (AIPE) of the standardized mean*

Description

A function to calculate the appropriate sample size for the standardized mean such that the width of the confidence interval is sufficiently narrow.

Usage

```
ss.aipe.sm(sm, width, conf.level = 0.95, assurance = NULL, certainty=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)
`certainty`	an alias for `assurance`
`...`	allows one to potentially include parameter values for inner functions

Value

`n`	the necessary sample size in order to achieve the desired degree of accuracy (i.e., the sufficiently narrow confidence interval)

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`); Keke Lai

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*, 532–574.

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 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.

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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.

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

`conf.limit.nct`, `ci.sm`

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, selected.n = NULL, assurance = NULL,
certainty=NULL, conf.level = 0.95, G = 10000, print.iter = TRUE,
detail = TRUE, ...)
```

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
`selected.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)
`certainty`	an alias for `assurance`
`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
`detail`	whether the user needs a detailed (`TRUE`) or brief (`FALSE`) report of the simulation results; the detailed report includes all the raw data in the simulations
`...`	allows one to potentially include parameter values for inner functions

Value

`sm.obs` vector of the observed standardized mean

`Full.Width` vector of the full confidence interval width

`Width.from.sm.obs.Lower` vector of the lower confidence interval width

`Width.from.sm.obs.Upper` vector of the upper confidence interval width

`Type.I.Error.Upper` iterations where a Type I error occurred on the upper end of the confidence interval

`Type.I.Error.Lower` iterations where a Type I error occurred on the lower end of the confidence interval

`Type.I.Error` iterations where a Type I error happens

`Lower.Limit` the lower limit of the obtained confidence interval

`Upper.Limit` the upper limit of the obtained confidence interval

`replications` number of replications of the simulation

`True.sm` the population standardized mean

`Estimated.sm` the estimated standardized mean

`Desired.Width` the desired full confidence interval width

`assurance` parameter to ensure that the obtained confidence interval width is narrower than the desired width with a specified degree of certainty

`Sample.Size` the sample size used in the simulation

`mean.full.width` mean width of the obtained full confidence intervals

`median.full.width` median width of the obtained full confidence intervals

`sd.full.width` standard deviation of the widths of the obtained full confidence intervals

`Pct.Width.obs.NARROWER.than.desired` percentage of the obtained full confidence interval widths that are narrower than the desired width

`mean.Width.from.sm.obs.Lower` mean lower width of the obtained confidence intervals

`mean.Width.from.sm.obs.Upper` mean upper width of the obtained confidence intervals

`Type.I.Error.Upper` Type I error rate from the upper side

`Type.I.Error.Lower` Type I error rate from the lower side

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>); Keke Lai

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*, 532–574.

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 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.

Kelley, K. (2007). Constructing confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20* (8), 1–24.

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.

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`

Examples

```
# Since 'true.sm' equals 'estimated.sm', this usage
# returns the results of a correctly specified situation.
# Note that 'G' should be large (10 is used to make the
# example run easily)
# Res.1 <- ss.aipe.sm.sensitivity(true.sm=10, estimated.sm=10,
# desired.width=.5, assurance=.95, conf.level=.95, G=10,
# print.iter=FALSE)

# Lists contained in Res.1.
# names(Res.1)

#Objects contained in the 'Results' lists.
# names(Res.1$Results)

#How many obtained full widths are narrower than the desired one?
# Res.1$Summary$Pct.Width.obs.NARROWER.than.desired

# True standardized mean difference is 10, but specified at 12.
# Change 'G' to some large number (e.g., G=20)
# Res.2 <- ss.aipe.sm.sensitivity(true.sm=10, estimated.sm=12,
# desired.width=.5, assurance=NULL, conf.level=.95, G=20)

# The effect of the misspecification on mean confidence intervals is:
# Res.2$Summary$mean.full.width
```

`ss.aipe.smd` *Sample size planning for the standardized mean difference from the Accuracy in Parameter Estimation (AIPE) perspective*

Description

A function to calculate the appropriate sample size for the standardized mean difference such that the expected value of the confidence interval is sufficiently narrow, optionally with a `degree.of.certainty`.

Usage

```
ss.aipe.smd(delta, conf.level, width, which.width="Full",
degree.of.certainty=NULL, assurance=NULL, certainty=NULL, ...)
```

Arguments

`delta`	the population value of the standardized mean difference
`conf.level`	the desired degree of confidence (i.e., 1-Type I error rate)
`width`	desired width of the specified (i.e., `Full`, `Lower`, and `Upper` widths) region of the confidence interval
`which.width`	the width that the `width` argument refers identifies the width of interest (i.e., `Full`, `Lower`, and `Upper` widths)
`degree.of.certainty`	parameter to ensure confidence interval width with a specified degree of certainty
`assurance`	an alias for `degree.of.certainty`
`certainty`	an alias for `degree.of.certainty`
`...`	for modifying parameters of functions this function calls upon

Value

Returns the necessary sample size *per group* in order to achieve the desired degree of accuracy (i.e., the sufficiently narrow confidence interval).

Warning

Finding sample size for lower and uppper confidence limits is approximate, but very close to being exact. The `pt()` function is limited to accurate values when the the noncentral parameter is less than 37.62.

Note

The function `ss.aipe.smd` is the preferred function, and is the one that is recommended for widespread use. The functions `ss.aipe.smd.lower`, `ss.aipe.smd.upper` and `ss.aipe.smd.full` are called from the `ss.aipe.smd` function.

Author(s)

Ken Kelley (University of Notre Dame; <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*, 532–574.

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 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.

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*, 258–287.

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.

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 where no significance tests?* (pp. 221-257). Mahwah, NJ: Lawrence Erlbaum.

See Also

`smd`, `smd.c`, `ci.smd`, `ci.smd.c`, `conf.limits.nct`, `power.t.test`, `ss.aipe.smd.lower`, `ss.aipe.smd.upper`, `ss.aipe.smd.full`

Examples

```
# ss.aipe.smd(delta=.5, conf.level=.95, width=.30)
# ss.aipe.smd(delta=.5, conf.level=.95, width=.30, degree.of.certainty=.8)
# ss.aipe.smd(delta=.5, conf.level=.95, width=.30, degree.of.certainty=.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, selected.n=NULL, assurance=NULL, certainty = NULL,
conf.level = 0.95, G = 10000, print.iter = TRUE, ...)
```

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
`selected.n`	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)
`certainty`	an alias for `assurance`
`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
`...`	for modifying parameters of functions this function calls

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 `certainty` parameter); see `ss.aipe.smd`)

Alternatively, one can specify `selected.n` to determine the results at a particular sample size (when doing this `estimated.delta` cannot be specified).

Value

`Results`	list of the results in `G`-length vector form
`Specifications`	specification of the function
`Summary`	summary measures of some important descriptive statistics
`d`	contained in `Results` list: vector of the observed `d` values
`Full.Width`	contained in `Results` list: vector of
`Width.from.d.Upper`	contained in `Results` list: vector of the observed upper widths of the confidence interval (upper limit minus observed standardized mean difference)

`Width.from.d.Lower`
: contained in `Results` list: vector of the observed lower widths of the confidence interval (standardized mean difference minus lower limit)

`Type.I.Error.Upper`
: contained in `Results` list: iterations where a Type I error occurred on the upper end of the confidence interval

`Type.I.Error.Lower`
: contained in `Results` list: iterations where a Type I error occurred on the lower end of the confidence interval

`Type.I.Error`
: contained in `Results` list: iterations where a Type I error occurred

`Upper.Limit`
: contained in `Results` list: vector of the obtained upper limits from the simulation

`Low.Limit`
: contained in `Results` list: vector of the obtained lower limits from the simulation

`replications`
: contained in `Specifications` list: number of generations (i.e., replication) of the simulation

`true.delta`
: contained in `Specifications` list: population value of the standardized mean difference

`estimated.delta`
: contained in `Specifications` list: value of the population (mis)specified for purposes of sample size planning

`desired.width`
: contained in `Specifications` list: desired full width of the confidence interval around the population standardized mean difference

`certainty`
: contained in `Specifications` list: desired degree of certainty that the obtained confidence interval width is less than the value specified

`n.j`
: contained in `Specifications` list: sample size per group given the specifications

`mean.full.width`
: contained in `Summary` list: mean width of the obtained confidence intervals

`median.full.width`
: contained in `Summary` list: median width of the obtained confidence intervals

`sd.full.width`
: contained in `Summary` list: standard deviation of the obtained confidence intervals

`Pct.Less.Desired`
: contained in `Summary` list: Percent of the confidence widths less than the width specified.

`mean.Width.from.d.Lower`
: contained in `Summary` list:mean width of the lower portion of the confidence interval (from d)

`mean.Width.from.d.Upper`
: contained in `Summary` list:mean width of the upper portion of the confidence interval (from d)

`Type.I.Error.Upper`
: contained in `Summary` list: Type I error rate from the upper side

`Type.I.Error.Lower`
: contained in `Summary` list: Type I error rate from the lower side

Note

Returns three lists, where each list has multiple components.

Author(s)

Ken Kelley (University of Notre Dame; <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*, 532–574.

Hedges, L. V. (1981). Distribution theory for Glass's Estimator of effect size and related estimators. *Journal of Educational Statistics, 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.

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`

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)
# Res.1 <- ss.aipe.smd.sensitivity(true.delta=.5, estimated.delta=.5,
# desired.width=.30, certainty=NULL, conf.level=.95, G=50,
# print.iter=FALSE)

# Lists contained in Res.1.
# names(Res.1)

#Objects contained in the 'Results' lists.
# names(Res.1$Results)

#Extract d from the Results list of Res.1.
# d <- Res.1$Results$d

# hist(d)

# Pull out summary measures
# Res.1$Summary

# 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, certainty=NULL, conf.level=.95, G=50,
# print.iter=FALSE)

# The effect of the misspecification on mean confidence intervals is:
# Res.2$Summary$mean.full.width

# 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, certainty=NULL, conf.level=.95, G=50,
# print.iter=FALSE)

# The effect of the misspecification on mean confidence intervals is:
# Res.3$Summary$mean.full.width
```

`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.k_X.without.k = NULL, K = NULL,
beta.k = NULL, width, which.width = "Full", sigma.Y = 1, sigma.X.k = 1,
RHO.XX = NULL, Rho.YX = NULL, which.predictor = NULL,
alpha.lower = NULL, alpha.upper = NULL, conf.level = .95,
degree.of.certainty = NULL, assurance=NULL, certainty=NULL,
Suppress.Statement = FALSE)
```

Arguments

`Rho2.Y_X` Population value of the squared multiple correlation coefficient

`Rho2.k_X.without.k`
Population value of the squared multiple correlation coefficient predicting the *k*th predictor variable from the remaining *p*-1 predictor variables

`K` the number of predictor variables

`beta.k` the regression coefficient for the *k*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.k` the population standard deviation of the kth 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)

`degree.of.certainty` 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.

`assurance` an alias for `degree.of.certainty`

`certainty` an alias for `degree.of.certainty`

`Suppress.Statement` `TRUE/FALSE` statement whether or not a sentence describing the situation defined is printed with the necessary sample size

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 MBESS but has a different naming scheme. See `ss.aipe.reg.coef` for more details.

Author(s)

Ken Kelley (University of Notre Dame; <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*, 305–321.

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

See Also

`ss.aipe.reg.coef.sensitivity`, `conf.limits.nct`, `ss.aipe.reg.coef`, `ss.aipe.rc`

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=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=1, RHO.XX=RHO.XX,
# Rho.YX=Rho.YX, which.predictor=1, conf.level=1-.05, degree.of.certainty=.85)
```

`ss.aipe.src.sensitivity` *Sensitivity analysis for sample size planing 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,
degree.of.certainty = NULL, assurance=NULL, certainty=NULL,
 G = 1000, print.iter = TRUE)
```

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 using `Estimated.Rho.YX` and `Estimated.RHO.XX`)
- `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)
- `degree.of.certainty` — degree of certainty that the obtained confidence interval will be sufficiently narrow
- `assurance` — an alias for `degree.of.certainty`
- `certainty` — an alias for `degree.of.certainty`
- `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

Details

Direct specification of `True.Rho.YX` and `True.RHO.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

- `Results` — a matrix containing the empirical results from each of the `G` replication of the simulation
- `Specifications` — a list of the input specifications and the required sample size
- `Summary.of.Results` — summary values for the results of the sensitivity analysis (simulation study) given the input specification

Note

Note that when `True.Rho.YX=Estimated.Rho.YX` and `True.RHO.XX=Estimated.RHO.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 MBESS for more details.

Author(s)

Ken Kelley (University of Notre Dame; <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*, 305–321.

See Also

`ss.aipe.reg.coef.sensitivity`, `ss.aipe.rc.sensitivity`,

`ss.aipe.reg.coef`, `ci.reg.coef`

`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 polynomial change models (currently only linear, that is, straight-line, change models)

Usage

```
ss.power.pcm(beta, tau, level.1.variance, frequency, duration, desired.power = NULL,
N = NULL, alpha.level = 0.05, standardized = TRUE, directional = FALSE)
```

Arguments

`beta`	the level two regression coefficient for the group by time (linear) interaction; where "X" is coded -.5 and .5 for the two groups.
`tau`	the true variance of the individuals' slopes
`level.1.variance`	level one variance
`frequency`	frequency of measurements per unit of time duration of the study in the particular units (e.g., age, hours, grade level, years, etc.)
`duration`	time in some number of units (e.g., years)
`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 slope is the unstandardized slope divided by the square root of tau, the variance of the unique effects for beta.
directional	should a one (TRUE) or two (FALSE) tailed test be performed.

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>)

References

Raudenbush, S. W., & X-F., Liu. (2001). Effects of study duration, frequency of observation, and sample size on power in studies of group differences in polynomial change. *Psychological Methods, 6*, 387–401.

Examples

```
# Example from Raudenbush and Liu (2001)
ss.power.pcm(beta=-.4, tau=.003, level.1.variance=.0262, frequency=2, duration=2,
desired.power=.80, alpha.level=.05, standardized=TRUE, directional=FALSE)
ss.power.pcm(beta=-.4, tau=.003, level.1.variance=.0262, frequency=2, duration=2,
N=238, alpha.level=.05, standardized=TRUE, directional=FALSE)


# The standardized effect size is obtained as beta/sqrt(tau): -.4/sqrt(.003) = -.0219.
# ss.power.pcm(beta=-.0219, tau=.003, level.1.variance=.0262, frequency=2, duration=2,
# desired.power=.80, alpha.level=.05, standardized=FALSE, directional=FALSE)
ss.power.pcm(beta=-.0219, tau=.003, level.1.variance=.0262, frequency=2, duration=2,
N=238, alpha.level=.05, standardized=FALSE, directional=FALSE)
```

---

ss.power.R2 *Function to plan sample size so that the test of the squared multiple correlation coefficient is sufficiently powerful.*

Description

Function for determining the necessary sample size for the test of the squared multiple correlation coefficient or for determining the statistical power given a specified sample size for the squared multiple correlation coefficient in models where the regressors are regarded as fixed.

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,
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)
`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)
`Print.Progress`	if the progress of the iterative procedure is printed to the screen as the iterations are occuring
`...`	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.

Value

`Sample.Size`	returns either `Necessary.Sample.Size` or `Specified.Sample.Size`, depending on if sample size is being determined for a desired degree of statistical power analysis or if statistical power is being determined given a specified sample size, respectively
`Actual.Power`	Actual power of the situation described

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 (University of Notre Dame; `<KKelley@ND.Edu>`)

See Also

`ss.aipe.R2`, `ss.power.reg.coef`, `conf.limits.ncf`

Examples

```
# ss.power.R2(Population.R2=.5, alpha.level=.05, desired.power=.85, p=5)
# ss.power.R2(Cohen.f2=1, alpha.level=.05, desired.power=.85, p=5)
# ss.power.R2(Population.R2=.5, Specified.N=15, alpha.level=.05,
# desired.power=.85, p=5)
# ss.power.R2(Cohen.f2=1, Specified.N=15, alpha.level=.05, desired.power=.85, p=5)
```

---

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.k = NULL, K = NULL,
desired.power = 0.85, alpha.level = 0.05, Directional = FALSE,
beta.k = NULL, sigma.X = NULL, sigma.Y = NULL,
Rho2.k_X.without.k = 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.k	population squared multiple correlation coefficient predicting the dependent variable (i.e., Y) from the K-1 predictor variables, where the one not used is the predictor of interest
K	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.k	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.k_X.without.k	population squared multiple correlation coefficient predicting the predictor variable of interest from the remaining K-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

`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.k`. See the examples section for several options.

Value

`Sample.Size` 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.NC.t` 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 (University of Notre Dame; <KKelley@ND.Edu>)

References

Maxwell, S. E. (2000). Sample size for multiple regression. *Psychological Methods, 4*, 434–458.

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

See Also

`ss.aipe.reg.coef`, `ss.power.R2`, `conf.limits.ncf`

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.k=0.7363697, K=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.k_X.without.k=0.3652136,
# beta.k=0.2700964, K=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, K=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.k=0.7363697,
# Specified.N=25, K=5, alpha.level=.05, Directional=FALSE)

# Power given a specified N and effect size.
# ss.power.rc(alpha.level=.05, Cohen.f2=0.2130898, K=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

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


# 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.k=R2.Maxwell.no.1, K=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

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)

which.predictor
: identifies the predictor of interest when RHO.XX and Rho.YX are specified

Specified.N
: sample size for which power should be evaluated

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

Sample.Size
: 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.NC.t
: 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 (University of Notre Dame; <KKelley@ND.Edu>)

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: 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. Alasuuta, J. Brannen, & L. Bickman (Eds.), *The Sage handbook of social research methods* (pp. 166–192). Newbury Park, CA: Sage.

Maxwell, S. E. (2000). Sample size for multiple regression. *Psychological Methods, 4*, 434–458.

See Also

ss.aipe.reg.coef, ss.power.R2, conf.limits.ncf

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

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


# 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
# in Cohen (1988)'s tables. The present procedure is correct and contains no
# rounding error
# in the application of the method.
# 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.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 = 0.05, power = 0.8)
```

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`	The Type I error rate.
`power`	The desired power.

Author(s)

Keke Lai (University of California - Merced)

`t.and.smd.conversion` *Conversion functions for noncentral t-distribution*

Description

Functions useful for converting a standardized mean difference to a noncentrality parameter, and vice versa.

Usage

```
lambda2delta(lambda, n.1, n.2)
delta2lambda(delta, n.1, n.2)
```

Arguments

`lambda`	noncentral value from a *t*-distribution
`delta`	population value of the standardized mean difference
`n.1`	sample size in group 1
`n.2`	sample size in group 2

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

Either the value of `delta` given `lambda` or `lambda` given `delta` (and the *per group* sample sizes).

Author(s)

Ken Kelley (University of Notre Dame; <`KKelley@ND.Edu`>)

See Also

`smd`, `ci.smd`, `ss.aipe.smd`

Examples

```
lambda2delta(lambda=2, n.1=113, n.2=113)
delta2lambda(delta=.266076, n.1=113, n.2=113)
```

`theta.2.Sigma.theta` *Compute the model-implied covariance matrix of an SEM model*

Description

Obtain the model-implied covariance matrix of manifest variables given a structural equation model and its model parameters

Usage

```
theta.2.Sigma.theta(model, theta, latent.vars)
```

Arguments

`model`	an RAM (reticular action model; e.g., McArdle & McDonald, 1984) specification of a structural equation model, and should be of class `mod`. The model is specified in the same manner as does the `sem` package; see `sem` and `specify.model` for detailed documentations about model specifications in the RAM notation.
`theta`	a vector containing the model parameters. The names of the elements in `theta` must be the same as the names of the model parameters specified in `model`.
`latent.vars`	a vector containing the names of the latent variables

Details

Part of the codes in this function are adapted from the function `sem` in the `sem` R package (Fox, 2006). This function uses the same notation to specify SEM models as does `sem`. Please refer to `sem` and the example below for more detailed documentation about model specification and the RAM notation. For technical discussion on how to obtain the model implied covariance matrix in the RAM notation given model parameters, see McArdle and McDonald (1984).

Value

`ram`	RAM matrix, including any rows generated for covariances among fixed exogenous variables; column 5 includes computed start values.
`t`	number of model parameters (i.e., the length of `theta`)
`m`	total number of variables (i.e., manifest variables plus latent variables)
`n`	number of observed variables
`all.vars`	the names of all variables (i.e., manifest plus latent)
`obs.vars`	the names of observed variables
`latent.vars`	the names of latent variables
`pars`	the names of model parameters
`P`	the P matrix in RAM notation
`A`	the A matrix in RAM notation
`Sigma.theta`	the model implied covariance matrix

Author(s)

Keke Lai (University of California–Merced)

References

Fox, J. (2006). Structural equation modeling with the sem package in R. *Structural Equation Modeling, 13*, 465–486.

Lai, K., & Kelley, K. (in press). Accuracy in parameter estimation for targeted effects in structural equation modeling: Sample size planning for narrow confidence intervals. *Psychological Methods*.

McArdle, J. J., & McDonald, R. P. (1984). Some algebraic properties of the reticular action model. *British Journal of Mathematical and Statistical Psychology, 37*, 234–251.

See Also

`sem`; `specify.model`

Examples

```
## Not run:
# to obtain the model implied covariance matrix of Model 2 in the simulation
# study in Lai and Kelley (2010), one can use the present function in the
# following manner.

library(sem)

# specify a model object in the RAM notation
model.2<-specify.model()
xi1 -> y1, lambda1, 1
xi1 -> y2, NA, 1
xi1 -> y3, lambda2, 1
xi1 -> y4, lambda3, 0.3
```

```
eta1 -> y4, lambda4, 1
eta1 -> y5, NA, 1
eta1 -> y6, lambda5, 1
eta1 -> y7, lambda6, 0.3
eta2 -> y6, lambda7, 0.3
eta2 -> y7, lambda8, 1
eta2 -> y8, NA, 1
eta2 -> y9, lambda9, 1
xi1 -> eta1, gamma11, 0.6
eta1 -> eta2, beta21, 0.6
xi1 <-> xi1, phi11, 0.49
eta1 <-> eta1, psi11, 0.3136
eta2 <-> eta2, psi22, 0.3136
y1 <-> y1, delta1, 0.51
y2 <-> y2, delta2, 0.51
y3 <-> y3, delta3, 0.51
y4 <-> y4, delta4, 0.2895
y5 <-> y5, delta5, 0.51
y6 <-> y6, delta6, 0.2895
y7 <-> y7, delta7, 0.2895
y8 <-> y8, delta8, 0.51
y9 <-> y9, delta9, 0.51


# to inspect the specified model
model.2

theta <- c(1, 1, 0.3, 1,1, 0.3, 0.3, 1, 1, 0.6, 0.6,
0.49, 0.3136, 0.3136, 0.51, 0.51, 0.51, 0.2895, 0.51, 0.2895, 0.2895, 0.51, 0.51)

names(theta) <- c("lambda1","lambda2","lambda3",
"lambda4","lambda5","lambda6","lambda7","lambda8","lambda9",
"gamma11", "beta21",
"phi11", "psi11", "psi22",
"delta1","delta2","delta3","delta4","delta5","delta6","delta7",
"delta8","delta9")

res<-theta.2.Sigma.theta(model=model.2, theta=theta,
latent.vars=c("xi1", "eta1","eta2"))

Sigma.theta <- res$Sigma.theta

## End(Not run)
```

---

transform_r.Z *Transform a correlation coefficient (r) into the scale of Fischer's* Z

Description

This function transform a correlation coefficient into the scale of Fischer's Z.

Usage

```
transform_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

returns a value on the scale of Fisher's *Z*, also called Fisher's emphZ\', from a given correlation value.

Author(s)

Ken Kelley (University of Notre Dame; <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.

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

See Also

transform_Z.r, ci.cc

Examples

```
# From Hays (1994, pp. 649--650)
transform_r.Z(.35)
```

---

transform_Z.r *Transform Fischer's Z into the scale of a correlation coefficient*

Description

A function to transform Fischer's *Z* into the scale of a correlation coefficient.

Usage

```
transform_Z.r(Z)
```

Arguments

`Z` Fisher's Z or Fisher's Z\' value.

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

returns a value on the scale of a correlation coefficieint from a value of Fisher's Z.

Author(s)

Ken Kelley (University of Notre Dame; <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.

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

See Also

`transform_r.Z`, `ci.cc`

Examples

```
# From Hays (1994, pp. 649--650)
transform_Z.r(0.3654438)
```

---

upsilon — *A function for estimating the mediation effect size as discussed in Lachowicz, Preacher, & Kelley (submitted).*

Description

This function implements the upsilon effect size statistic for general mediation from Lachowicz, Preacher, & Kelley (submitted).

Usage

```
upsilon(data, x, m, y, covs = NULL, parallel.med = NULL, seq.med = NULL)
```

Arguments

`data`	`data` is the data that contains the variable that are to be used in the mediation model.
`x`	The `x` is the independent variable or variables (listed in a vector) that are included in `data`. That is, this is a string identifying the `x` variables.
`m`	The `m` is the mediator variable or variables (listed in a vector) that are included in `data`. That is, this is a string identifying the `m` variables.
`y`	The `y` is the outcome or dependent variable or variables (listed in a vector) that are included in `data`. That is, this is a string identifying the `y` variables.
`covs`	Covariates to include in the mediation model are identified in `covs`. That is, this is a string identifying the `covs` (covariates) to inlucde.
`parallel.med`	Identifies which of the mediators are parallel mediators. Not that the parallel mediators listed in `parallel.med` are also included in `m`.
`seq.med`	Identifies which of the mediators are sequential and in the order in a `list`. That is, `seq.med` identifying the sequential mediators and their order. The order is such that `list('m2~m1','m3~m2')` `m2` is caused by `m1` and `m3` is caused by `m2`.

Details

See the examples below for example applications of the function.

Value

`IE`	is the value of the indirect effect for the path noted. Note that the total, specific, and unconditional paths will be produced (depending on the model)
`Upsilon`	is the value of the effect size for the path noted. Note that the total, specific, unconditional and unique paths will be produced (depending on the model)

Note

Note that this function overcomes some limitations of other effects for mediation models, such as those discussed in Preacher and Kelley (2012) and Wen and Fan (2015).

Author(s)

Lachowicz Mark J. Lachowicz (Vanderbilt University; `<Mark.J.Lachowicz@Vanderbilt.edu>`)

References

Lachowicz, M. J., Preacher, K. J., & Kelley, K. (submitted). A novel measure of effect size for mediation analysis. Submited for publication.

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

Wen, Z., & Fan, X. (2015). Monotonicity of effect sizes: Questioning kappa-squared as mediation effect size measure. *Psychological Methods*, *20*, 193–203.

See Also

mediation

Examples

```
## Not run:
# To generate the multivariate data for the examples.
require(MASS)

# Generate data for example 1 and 2.
X<-matrix(c(1,.4,.2,.3,
            .4,1,.4,.1,
            .2,.4,1,.3,
            .3,.1,.3,1),4,4,byrow=TRUE)
data<-mvrnorm(500,c(0,0,0,0),X,empirical=TRUE)
colnames(data)<-c('x','m1','m2','y')

data <- as.data.frame(data)

#3 Example 1; three variable mediation; simple mediation model.
x<-'x'
m<-'m1'
y<-'y'

upsilon(data,x,m,y)

## Example 2; four variable mediation; 2 mediators
data<-mvrnorm(100,c(0,0,0,0),X,empirical=TRUE)
colnames(data)<-c('x1','x2','m','y')
data <- as.data.frame(data)

x <- c('x1','x2')
m <- 'm'
y <- 'y'

B <- upsilon(data,x,m,y)
B


# Generate data for example 3.
X1<-matrix(c(1,.4,.3,.3,.1,
             .4,1,-.2,.1,.2,
             .3,-.2,1,-.1,.3,
             .3,.1,-.1,1,.4,
             .1,.2,.3,.4,1),5,5,byrow=TRUE)

data<-mvrnorm(100,c(0,0,0,0,0),X1,empirical=TRUE)
colnames(data)<-c('x1','x2','x3','m','y')
data <- as.data.frame(data)

## Example 3; 3 predictors, 1 mediator, 1 outcome
x <- c('x1','x2','x3')
```

```
m <- 'm'
y <- 'y'

C<-upsilon(data,x,m,y)
C

# Generate data for example 4.
X1<-matrix(c(1,.2,.3,.3,.1,
             .2,1,-.2,.3,.2,
             .3,-.2,1,.3,.3,
             .3,.3,.3,1,.4,
             .1,.2,.3,.4,1),5,5,byrow=TRUE)

data<-mvrnorm(100,c(0,0,0,0,0),X1,empirical=TRUE)
colnames(data)<-c('x1','x2','x3','m','y')
data <- as.data.frame(data)

## Example 4; 2 predictors, 1 mediator, 1 outcome, 1 covariate
x <- c('x1','x2')
m <- 'm'
y <- 'y'
cov <-'x3'

upsilon(data,x,m,y,covs=cov)

# Generate data for example 5.
data<-mvrnorm(100,c(0,0,0,0),X,empirical=TRUE)
colnames(data)<-c('x','m1','m2','y')
data<-as.data.frame(data)

## Example 5; 1 predictor, 1 outcome, 2 parallel mediators
x <- 'x'
m <- c('m1','m2')
y <- 'y'
meds <- c('m1+m2')

upsilon(data,x,m,y,parallel.med=meds)

# Generate data for example 6.
data<-mvrnorm(100,c(0,0,0,0,0),X1,empirical=TRUE)
colnames(data)<-c('x','m1','m2','m3','y')
data<-as.data.frame(data)

## Example 6, 1 predictor, 1 outcome, 3 parallel mediators
x <- 'x'
m <- c('m1','m2','m3')
y <- 'y'
meds <- c('m1+m2+m3')

upsilon(data,x,m,y,parallel.med=meds)

# Generate data for example 7.
data<-mvrnorm(100,c(0,0,0,0,0),X1,empirical=TRUE)
```

```
colnames(data)<-c('x1','x2','m1','m2','y')
data <- as.data.frame(data)

# Example 7; 2 predictors, 1 outcome, 2 parallel mediators
x <- c('x1','x2')
m <- c('m1','m2')
y <- 'y'
parallel.med <- c('m1+m2')

upsilon(data,x,m,y,parallel.med=parallel.med)

# Generate data for example 8.
data<-mvrnorm(100,c(0,0,0,0),X,empirical=TRUE)
colnames(data)<-c('x','m1','m2','y')
data<-as.data.frame(data)

## Example 8; 2 mediators, serial
x <- 'x'
m <- c('m1','m2')
y <- 'y'
seq.med <- c('m2~m1')

upsilon(data,x,m,y,seq.med=seq.med)

# Generate data for example 9.
X1<-matrix(c(1,.4,.3,.3,.1,
             .4,1,-.1,.3,.2,
             .3,-.1,1,.3,.3,
             .3,.3,.3,1,.4,
             .1,.2,.3,.4,1),5,5,byrow=TRUE)

data<-mvrnorm(100,c(0,0,0,0,0),X1,empirical=TRUE)
colnames(data)<-c('x','m1','m2','m3','y')
data <- as.data.frame(data)

# Example 9; 3 sequential mediators
x <- c('x')
m <- c('m1','m2','m3')
y <- c('y')
seq.med <- list('m2~m1','m3~m2')

upsilon(data,x,m,y,seq.med=seq.med)

## End(Not run)
```

Variance.R2 — *Variance of squared multiple correlation coefficient*

Description

Function to determine the variance of the squared multiple correlation coefficient given the population squared multiple correlation coefficient, sample size, and the number of predictors.

Usage

```
Variance.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

Returns the variance of the of the squared multiple correlation coefficient.

Note

Uses package `gsl` and its `hyperg_2F1` function.

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`)

References

Stuart, A., Ord, J. K., & Arnold, S. (1999). *Kendall's advanced theory of statistics: Classical inference and the linear model* (Volume 2A, 2nd Edition). New York, NY: Oxford University Press.

See Also

`Expected.R2`, `ci.R2`, `ss.aipe.R2`

Examples

```
# library(gsl)
# Variance.R2(.5, 10, 5)
# Variance.R2(.5, 25, 5)
# Variance.R2(.5, 50, 5)
# Variance.R2(.5, 100, 5)
```

`verify.ss.aipe.R2` *Internal MBESS function for verifying the sample size in ss.aipe.R2*

Description

Internal function called upon by `ss.aipe.R2` when `verify.ss=TRUE`. This function then calls upon `ss.aipe.R2.sensitivity` for the simulation study.

Usage

```
verify.ss.aipe.R2(Population.R2 = NULL, conf.level = 0.95, width = NULL,
Random.Predictors = TRUE, which.width = "Full", p = NULL, n = NULL,
degree.of.certainty = NULL, g = 500, G = 10000, print.iter=FALSE, ...)
```

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
`n`	starting sample size (i.e., from `ss.aipe.R2`)
`degree.of.certainty`	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)
`g`	simulations for the preliminary sample size (much smaller than `G`)
`G`	number of replications for the actual Monte Carlo simulation (should be large)
`print.iter`	specify whether or not the internal iterations should be printed
`...`	additional arguments passed to internal functions

Details

This function is internal to MBESS and is called upon when `verify.ss=TRUE` in the `ss.aipe.R2` function. Although users can use `verify.ss.aipe.R2` directly, it is not recommended.

Value

Returns the exact (provided `G` is large enough) sample size necessary to satisfy the condidtions specified.

Author(s)

Ken Kelley (University of Notre Dame; `<KKelley@ND.Edu>`)

vit *Visualize individual trajectories*

Description

A function to help visualize individual trajectories in a longitudinal (i.e., analysis of change) context.

Usage

```
vit(id = "", occasion = "", score = "", Data = NULL, group = NULL,
subset.ids = NULL, pct.rand = NULL, number.rand = NULL,
All.in.One = TRUE, ylab = NULL, xlab = NULL, same.scales = TRUE,
plot.points = TRUE, save.pdf = FALSE, save.eps = FALSE,
 save.jpg = FALSE, file = "", layout = c(3, 3), col = NULL,
 pch = 16, cex = 0.7, ...)
```

Arguments

`id`	string variable of the column name of id
`occasion`	string variable of the column name of time variable
`score`	string variable of the column name where the score (i.e., dependent variable) is located
`Data`	data set with named column variables (see above)
`group`	if plotting parameters should be conditional on group membership
`subset.ids`	id values for a selected subset of individuals
`pct.rand`	percentage of random trajectories to be plotted
`number.rand`	number of random trajectories to be plotted
`All.in.One`	should trajectories be in a single or multiple plots
`ylab`	label for the ordinate (i.e., y-axis; see par)
`xlab`	label for the abscissa (i.e., x-axis; see par)
`same.scales`	should the y-axes have the same scales
`plot.points`	should the points be plotted
`save.pdf`	save a pdf file
`save.eps`	save a postscript file
`save.jpg`	save a jpg file
`file`	file name and file path for the graph(s) to save, if `file=""` a file would be saved in the current working directory
`layout`	define the per-page layout when `All.in.One=FALSE`
`col`	color(s) of the line(s) and points
`pch`	plotting character(s); see par
`cex`	size of the points (1 is the R default; see par)
`...`	optional plotting specifications

Details

This function makes visualizing individual trajectories simple. Data should be in the "univariate format" (i.e., the same format as lmer and nlme data.)

Value

Returns a plot of individual trajectories with the specifications provided.

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>) and Po-Ju Wu (Indiana University)

See Also

par, nlme, vit.fitted,

Examples

```
## Not run:
data(Gardner.LD)

# Although many options are possible, a simple call to
# 'vit' is of the form:
# vit(id="ID", occasion= "Trial", score= "Score", Data=Gardner.LD)

# Now color is conditional on group membership.
# vit(id="ID", occasion= "Trial", score="Score", Data=Gardner.LD,
# group="Group")

# Now randomly selects 50
# vit(id="ID", occasion= "Trial", score="Score", Data=Gardner.LD,
# pct.rand=50, group="Group")

# Specified individuals are plotted (by group)
# vit(id="ID", occasion= "Trial", score="Score", Data=Gardner.LD,
# subset.ids=c(1, 4, 8, 13, 17, 21), group="Group")

# Now colors for groups are changed .
# vit(id="ID", occasion= "Trial", score="Score", Data=Gardner.LD,
# group="Group",subset.ids=c(1, 4, 8, 13, 17, 21), col=c("Green", "Blue"))

# Now each individual specified is plotted separately.
# vit(id="ID", occasion= "Trial", score="Score", Data=Gardner.LD,
# group="Group",subset.ids=c(1, 4, 8, 13, 17, 21), col=c("Green", "Blue"),
# All.in.One=FALSE)

## End(Not run)
```

`vit.fitted` *Visualize individual trajectories with fitted curve and quality of fit*

Description

A function to help visualize individual trajectories in a longitudinal (i.e., analysis of change) context with fitted curve and quality of fit after analyzing the data with lme, lmer, or nlme function.

Usage

```
vit.fitted(fit.Model, layout = c(3, 3), ylab = "", xlab = "",
pct.rand = NULL, number.rand = NULL, subset.ids = NULL,
same.scales = TRUE, save.pdf = FALSE, save.eps = FALSE,
save.jpg = FALSE, file = "", ...)
```

Arguments

`fit.Model`	lme, nlme object produced by nlme package or lmer object produced by lme4 package
`layout`	define the per-page layout when `All.in.One=FALSE`
`ylab`	label for the ordinate (i.e., y-axis; see par)
`xlab`	label for the abscissa (i.e., x-axis; see par)
`pct.rand`	percentage of random trajectories to be plotted
`number.rand`	number of random trajectories to be plotted
`subset.ids`	id values for a selected subset of individuals to be plotted
`same.scales`	should the y-axes have the same scales
`save.pdf`	save a pdf file
`save.eps`	save a postscript file
`save.jpg`	save a jpg file
`file`	file name and file path for the graph(s) to save, if `file=""` a file would be saved in the current working directory
`...`	optional plotting specifications

Details

This function uses the fitted model from nlme and lme functions in nlme package, and lmer function in lme4 package. It returns a set of plots of individual observed data, the fitted curves and the quality of fit.

Author(s)

Ken Kelley (University of Notre Dame; <KKelley@ND.Edu>) and Po-Ju Wu (Indiana University; <pojwu@indiana.edu>)

See Also

par, nlme, lme4, lme, lmer, vit.fitted

Examples

```
## Not run:
# Note that the following example works fine in R (<2.7.0), but not in
# the development version of R-2.7.0 (the cause can be either in this
# function or in the R program)

# data(Gardner.LD)
# library(nlme)
# Full.grouped.Gardner.LD <- groupedData(Score ~ Trial|ID, data=Gardner.LD, order.groups=FALSE)

# Examination of the plot reveals that the logistic change model does not adequately describe
# the trajectories of individuals 6 and 19 (a negative exponential change model would be
# more appropriate). Thus we remove these two subjects.
# grouped.Gardner.LD <- Full.grouped.Gardner.LD[!(Full.grouped.Gardner.LD["ID"]==6 |
#   Full.grouped.Gardner.LD["ID"]==19),]

# G.L.nlsList<- nlsList(SSlogis,grouped.Gardner.LD)
# G.L.nlme <- nlme(G.L.nlsList)
# to visualize individual trajectories:  vit.fitted(G.L.nlme)
# plot 50 percent random trajectories:  vit.fitted(G.L.nlme, pct.rand = 50)

## End(Not run)
```

Index