The MBESS Package

June 2, 2007

Title Methods for the Behavioral, Educational, and Social Sciences

Version 0.0.9

Date 2007-5-31

Author Ken Kelley

Maintainer Ken Kelley <KKIII@Indiana.Edu>

Depends R (>= 2.5.0)

Imports MASS, gsl

Suggests MASS, nlme

Description MBESS implements methods that are especially useful to researchers working within the behavioral, educational, and social sciences (both substantive researchers and methodologists). Many of the methods contained within MBESS are applicable to quantitative research in general.

License General Public License 2.0 or more recent. See file LICENSE for more information.


R topics documented:

Expected.R2 ......................................................... 3
F.and.R2.Noncentral.Conversion ................................. 4
Gardner.LD ......................................................... 5
HS.data .......................................................... 6
MBESS ........................................................... 8
Variance.R2 ......................................................... 8
aipe.smd .......................................................... 10
ci.R ................................................................. 11
ci.R2 ............................................................... 13
ci.c ................................................................. 15
ci.c.ancova ......................................................... 17
ci.cv ................................................................. 19
ci.pvaf ............................................................ 20
2

R topics documented:

- ci.rc .......................................................... 22
- ci.reg.coef .................................................... 24
- ci.sc ........................................................... 26
- ci.sc.ancova .................................................. 27
- ci.sm ........................................................... 29
- ci.smd .......................................................... 31
- ci.smd.c ....................................................... 32
- ci.snr ........................................................... 34
- ci.src ........................................................... 36
- ci.srsnr ........................................................ 38
- conf.limits.nc.chisq .......................................... 39
- conf.limits.ncf ............................................... 41
- conf.limits.net.M1 .......................................... 42
- conf.limits.net.M2 .......................................... 43
- conf.limits.net.M3 .......................................... 44
- conf.limits.nct ............................................... 46
- cor2cov ......................................................... 48
- cv ............................................................... 48
- intr.plot.2d .................................................. 49
- intr.plot ....................................................... 51
- prof.salary .................................................... 53
- s.u .............................................................. 54
- signal.to.noise.R2 .......................................... 55
- smd ............................................................. 56
- smd.c .......................................................... 58
- ss.aipe.R2 ..................................................... 60
- ss.aipe.R2.sensitivity ........................................ 62
- ss.aipe.c ....................................................... 65
- ss.aipe.c.ancova ............................................. 66
- ss.aipe.c.sensitivity.ancova ................................ 67
- ss.aipe.cv ...................................................... 70
- ss.aipe.cv.sensitivity ........................................ 71
- ss.aipe.rc ...................................................... 73
- ss.aipe.rc.sensitivity ........................................ 75
- ss.aipe.reg.coef ............................................. 77
- ss.aipe.reg.coef.sensitivity ................................ 79
- ss.aipe.sc ...................................................... 81
- ss.aipe.sc.sensitivity ........................................ 82
- ss.aipe.sm ...................................................... 85
- ss.aipe.sm.sensitivity ....................................... 86
- ss.aipe.smd .................................................... 89
- ss.aipe.smd.sensitivity ...................................... 91
- ss.aipe.src .................................................... 94
- ss.aipe.src.sensitivity ....................................... 97
- ss.power.R2 ................................................... 99
- ss.power.rc .................................................. 100
- ss.power.reg.coef ........................................... 103

0.1 Warning .................................................... 96

- ss.aipe.src.sensitivity ....................................... 97
- ss.power.R2 ................................................... 99
- ss.power.rc .................................................. 100
- ss.power.reg.coef ........................................... 103
Expected.R2


t.and.smd.conversion .................................................. 107
verify.ss.aipe.R2 .......................................................... 108
vit .................................................. 109
vit.fitted .................................................. 111

Index

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 and 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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References

Mathematical statistics, 29, 201–211.
See Also

'ss.aipe.R2', 'ci.R2', 'Variance.R2'

Examples

Expected.R2(.5, 10, 5)
Expected.R2(.5, 25, 5)
Expected.R2(.5, 50, 5)
Expected.R2(.5, 100, 5)
Expected.R2(.5, 1000, 5)
Expected.R2(.5, 10000, 5)


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 for 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 (Indiana University; ⟨KKIII@Indiana.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) where presented the letters S, L, N, and D with probabilities .70, .10, .10, and .10, respectively. The other 12 participants (Group 2) where 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 where 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


References


---

**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 on
- **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
Details

Holzinger and Swineford (1939) data is a widely-cited one, but almost always only a part of the dataset, the scores for Grant-White School, were 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 substitute 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, paragraf, 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

References

---

**MBESS**

*Methods for the Behavioral, Educational, and Social Sciences*

Description

MBESS implements methods that are especially applicable to researchers working within the behavioral, educational, and social sciences. Many of the functions are also applicable to disciplines related to the behavioral, educational, and social sciences.

Details

- **Package:** MBESS
- **Type:** Package
- **Version:** 0.0.9
- **Date:** 2007-5-31
- **License:** General Public License 2 or more recent.

Please read the manual and visit the corresponding web site [http://www.indiana.edu/~kenkel/mbess](http://www.indiana.edu/~kenkel/mbess) 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 <KKIII@Indiana.Edu>; [http://www.indiana.edu/ kenkel](http://www.indiana.edu/ kenkel)

Maintainer: Ken Kelley <KKIII@Indiana.Edu>; [http://www.indiana.edu/ kenkel/mbess](http://www.indiana.edu/ kenkel/mbess)

---

**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.
Variance.R2

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 squared multiple correlation coefficient.

Note

Uses package gsl and its hyperg_2F1 function.

Author(s)

Ken Kelley (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References


See Also

'Expected.R2', 'ci.R2', 'ss.aipe.R2'

Examples

Variance.R2(.5, 10, 5)
Variance.R2(.5, 25, 5)
Variance.R2(.5, 50, 5)
Variance.R2(.5, 100, 5)
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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)
References


See Also

'ss.aipe.smd'

---

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

**Description**

Function to obtain the exact confidence interval for the 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 most 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 Algina 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
Returns the confidence limits of the multiple correlation coefficient:

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 Lower.Conf.Limit.R

Author(s)
Ken Kelley (Indiana University, <KKIII@Indiana.edu>)
ci.R2

References


See Also

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

---

ci.R2

Confidence intervals for the squared multiple correlation coefficient

Description

A function to calculate the exact confidence interval for the 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 multiple correlation coefficient
df.1 numerator degrees of freedom
df.2 denominator degrees of freedom
conf.level confidence interval coverage; 1-Type I error rate
Random.Predictors whether or not the predictor variables are random or fixed (random is default)
F.value obtained F-value
N sample size
p number of predictors
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 most applications in the behavioral, educational, and social sciences 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

Author(s)
Ken Kelley (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References
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.

Usage

```r
ci.c(means = NULL, error.variance = 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)
- **error.variance**: the common variance of the error (i.e., the mean square error)

See Also

'ss.aipe.R2', 'conf.limits.nct'

Examples

```r
# 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)
```
c.weights the contrast weights (the sum of the contrast weights should be zero)
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 jth 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

Details

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

Author(s)

Ken Kelley (Indiana University, <KKIII@Indiana.edu>)

References


See Also

'conf.limits.nct', 'ci.sc', 'ci.src', 'ci.smd', 'ci.smd.c', 'ci.sm'
Examples

ci.c(means=c(2, 4, 9, 13), error.variance=1, c.weights=c(1, -1, -1, 1),
n=c(3, 3, 3, 3), N=12, conf.level=.95)

ci.c(means=c(2, 4, 9, 13), error.variance=1, c.weights=c(1, -1, -1, 1),
n=c(3, 3, 3, 3), N=12, conf.level=.95)

ci.c(means=c(1.6, 0), error.variance=1, 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
# confidence interval for the difference in blood pressure between subjects
# received drug treatment and those 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), error.variance=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**

```r
ci.c.ancova(Psi, means, error.var.ancova = NULL, c.weights, n, x.bar, SSwithin.x,
conf.level = 0.95, ...)
```

**Arguments**

- **Psi**: the unstandardized contrast of adjusted means
- **means**: the vector that contains the adjusted mean of each group
- **error.var.ancova**: the error variance obtained from the ANCOVA summary table; i.e., mean square within in the ANCOVA table
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
x.bar    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
lower.limit  the upper confidence limit of the (unstandardized) ANCOVA contrast

Note

Be sure to use the error variance and not its square root (i.e., the standard deviation of the errors).
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 (Indiana University; ⟨LaiK@Indiana.Edu⟩)

References

Maxwell, S. E., & Delaney, H. D. (2004). Designing experiments and analyzing data: A model

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:

```r
ci.c.ancova(means=c(7.5, 12, 14), error.var.ancova=29, c.weights=c(1, -1, 0),
            n=10, x.bar=c(17, 17.7, 17.4), SSwithin.x=313.37)
```

---

**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**

```r
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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References


Examples

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

# using individual variables
ci.cv(mean=mean.X, sd=sd.X, n=N, alpha.lower=.025, alpha.upper=.025, conf.level=NULL)
# using data
ci.cv(data=X, conf.level=.95)
# using coefficient of variation
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**

```r
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 exact confidence interval for the proportion of variance of the dependent variable accounted for by knowing group status in a fixed factor analysis of variance.


The lower confidence limit for the proportion of variance accounted for in the deviation by group status.


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 (Indiana University;<KKIII@Indiana.Edu>)

References


## 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

```r
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 kth predictor variable
- `SE.b.k`: standard error for the kth 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

---

## See Also

'conf.limits.ncf'

## Examples

```r
## 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)
```
ci.rc

R2.k_X.without.k
the squared multiple correlation coefficient predicting the kth 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 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

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/FALSE statement 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
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 (Indiana University; KKIII@Indiana.Edu)

References


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**

```r
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`, 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
ci.reg.coef

**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/FALSE statement 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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

**References**


See Also

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

---

**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**

```r
ci.sc(means = NULL, error.variance = 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)
- **error.variance**: the common variance of the error (i.e., the mean square error)
- **c.weights**: the contrast weights (the sum of the contrast weights should be zero)
- **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 $j$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 error variance and not its square root (i.e., the standard deviation of the errors).

Author(s)

Ken Kelley (Indiana University, <KKIII@Indiana.edu>)

References


See Also

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

Examples

```r
> ci.sc(means=c(2, 4, 9, 13), error.variance=1, c.weights=c(1, -1, -1, 1), n=c(3, 3, 3, 3), N=12, conf.level=.95)
> ci.sc(means=c(2, 4, 9, 13), error.variance=1, c.weights=c(1, -1, -1, 1), n=c(3, 3, 3, 3), N=12, conf.level=.95)
> ci.sc(means=c(1.6, 0), error.variance=1, c.weights=c(1, -1), n=c(10, 10), N=20, conf.level=.95)
```

---

**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, means=NULL, s.anova = NULL, s.ancova, standardizer = "s.ancova", c.weights, n, x.bar, SSwithin.x, conf.level = 0.95)

Arguments

Psi     unstandardized contrast of adjusted means
means   the vector that contains the adjusted mean of each group
s.anova the error standard deviation of the ANOVA model
s.ancova the error standard deviation of the ANCOVA model
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
n       either a single number that indicates the sample size per group, or a vector that contains the sample size of each group
x.bar   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.limit.upper the upper confidence limit of the standardized contrast

Note

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

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 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.

The argument to be assigned to standardizer must be either "s.ancova" or "s.anova".

Author(s)

Keke Lai (Indiana University; LaiK@Indiana.Edu)
References

See Also
ci.c.ancova, ci.sc

Examples

```r
# 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 obtained the confidence interval for the standardized adjusted
# mean difference between group 1 and 2, using the ANCOVA error standard
development:
ci.sc.ancova(means=c(7.5, 12, 14), s.ancova=29, c.weights=c(1,-1,0),
n=10, x.bar=c(17, 17.7, 17.4), SSwithin.x=313.37)
```

---

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

**Description**

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

**Usage**

```r
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 distribution. The confidence interval for noncentral t parameter can be obtained from function ‘conf.limits.nct’ in MBESS.

Value

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower.Conf.Limit.Standardized.Mean</td>
<td>lower confidence limit of the standardized mean</td>
</tr>
<tr>
<td>Standardized.Mean</td>
<td>standardized mean</td>
</tr>
<tr>
<td>Upper.Conf.Limit.Standardized.Mean</td>
<td>upper confidence limit of the standardized mean</td>
</tr>
</tbody>
</table>

Note

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

Author(s)

Ken Kelley (Indiana University;<KKIII@Indiana.Edu>)

References


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 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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References

See Also
'smd.c', 'smd', 'ci.smd.c', 'conf.limits.nct'

Examples

```r
# Examples Steiger and Fouladi 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**

```r
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, ...)
```
ci.smd.c

**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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

**References**


See Also

'smd.c', 'smd', 'ci.smd', 'conf.limits.nct'

Examples

```r
ci.smd.c(smd.c=.5, n.C=100, conf.level=.95, n.E=100)
```

---

**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

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

Value

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


lower limit for 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 (Indiana University;<KKIII@Indiana.Edu>)

References


See Also

'ci.srsnr', 'conf.limits.ncf'

Examples

```r
## 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.src(F.value=11.221, df.1=4, df.2=50, N=55, alpha.lower=.02, alpha.upper=.03)

---

**Confidence Interval for a Standardized Regression Coefficient**

**Description**

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

**Usage**

```r
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 standardized 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 $k$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.k`: the squared multiple correlation coefficient predicting $Y$ from the $p-1$ 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
- `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/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 unstanardized 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 (Indiana University; KKIII@Indiana.Edu)

References

coefficients that are accuracy, not simply significant. Psychological Methods, 8, 305–321.

of sample size planning with applications to Multiple Regression. Handbook of Social Research


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

```r
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 obtained from function `conf.limits.ncf` in MBESS.

Value

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

Author(s)
Ken Kelley (Indiana University;<KKIII@Indiana.Edu>)

References


See Also
'ci.snr', 'conf.limits.ncf'

Examples
```r
## 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 example continued to be
## used in Venables (1975), Fleishman (1980), and Steiger (2004). If one wants to calculate
## the exact confidence interval for square root of the signal-to-noise ratio of that example,
## this function can be used.

ci.srsnr(F.value=11.221, df.1=4, df.2=50, N=55)
ci.srsnr(F.value=11.221, df.1=4, df.2=50, N=55, conf.level=.90)
ci.srsnr(F.value=11.221, df.1=4, df.2=50, N=55, alpha.lower=.02, alpha.upper=.03)
```

Description
Function to determine the noncentral parameter that leads to the observed Chi.Square-value, so that a confidence interval for 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
```r
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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

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)
**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**

```r
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)

**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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

See Also
'ss.aipe.R2', 'ci.R2', 'conf.limits.nct'

Examples

```r
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.M1 Confidence limits for a noncentrality parameter from a t-distribution (Method 1 of 3)

Description
Largely internal function to determine the noncentrality parameters necessary to form a confidence interval for the population noncentrality parameter and related parameters. Method 1 uses the optimize function to determine the critical values. This function requires the ncp to be positive, but the function that should be used conf.limits.nct does not.

Usage

```r
conf.limits.nct.M1(ncp, df, conf.level = .95, alpha.lower = NULL, alpha.upper = NULL, min.ncp = -3 * ncp, max.ncp = 3 * ncp, 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`)
- `min.ncp` lower noncentral parameter from which to start the search process
- `max.ncp` lower noncentral parameter from which to start the search process
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

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

Author(s)

Ken Kelley (Indiana University; ⟨KKIII@Indiana.Edu⟩)

See Also

conf.limits.nct, optimize

Description

Largely internal function to determine the noncentrality parameters necessary to form a confidence interval around the population noncentrality parameter and related parameters. Method 2 uses the nlm function to determine the critical values. This function requires the ncp to be positive, but the function that should be used conf.limits.nct does not.

Usage

conf.limits.nct.M2(ncp = ncp, df = df, conf.level = .95, alpha.lower = NULL, alpha.upper = NULL, 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)
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

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

Author(s)

Ken Kelley (Indiana University; ⟨KKIII@Indiana.Edu⟩)

See Also

conf.limits.nct, nlm

---

Description

Largely internal function to determine the noncentrality parameters necessary to form a confidence interval around the population noncentrality parameter and related parameters. Method 3 uses an iterative scheme in order to determine the critical values. This function requires the ncp to be positive, but the function that should be used conf.limits.nct does not.
Usage

conf.limits.nct.M3(ncp, df, conf.level = .95, alpha.lower = NULL, alpha.upper = NULL, tol = 1e-09, sup.int.warns = TRUE, max.steps = 2500, ...)

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)
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
max.steps maximum number of iterations when finding the lower and upper confidence limit
... allows one to potentially include parameter values for inner functions

Value

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

Note

This code was adapted from code written by Ken Kelley and Joseph R. Rausch (University of Notre Dame; JRausch@ND.Edu), which was adapted from code written by Michael Smithson (Australian National University; Michael.Smithson@ANU.Edu.AU).

Author(s)

Ken Kelley (Indiana University; KKIII@Indiana.Edu)

See Also

conf.limits.nct
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. Due to the difficulties in estimating the necessary values, three different methods are implemented within the present function (conf.limits.nct.M1, conf.limits.nct.M2, and conf.limits.nct.M3) and the best set of results are taken.

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, method = "all", ...)

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
method  which of the three methods should be used: "all", "1", "2", "3" ("all" is default).
...  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 curves, respectively. 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.

See the documentation and code for each of the three methods (if interested). Each of the three methods should reach the same values for the confidence limits. However, due to the iterative
nature of the functions, one function may arrive at a more optimal solution. Furthermore, in some situations one (or more) functions could fail to find the optimum values, which necessitates the use of multiple methods to (essentially) ensure that optimal values are found.

Value

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower.Limit</td>
<td>Value of the distribution with Lower.Limit noncentral value that has at its specified quantile F.value</td>
</tr>
<tr>
<td>Prob.Less.Lower</td>
<td>Proportion of the distribution beyond (i.e., less than) Lower.Limit</td>
</tr>
<tr>
<td>Upper.Limit</td>
<td>Value of the distribution with Upper.Limit noncentral value that has at its specified quantile F.value</td>
</tr>
<tr>
<td>Prob.Greater.Upper</td>
<td>Proportion of the distribution beyond (i.e., larger than) Upper.Limit</td>
</tr>
</tbody>
</table>

Warning

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

Author(s)

Ken Kelley (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References


See Also

'pt', 'qt', 'ci.smd', 'ci.smd.c', 'ss.aipe', 'conf.limits.ncf', 'conf.limits.nc.chisq'

Examples

```r
# 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 nonsymmetric confidence intervals # can be formed:
```
**conf.limits.nct**

(ncp=2.83, df=126, alpha.lower=.01, alpha.upper=.04, conf.level=NULL)

---

**cor2cov**  
*Correlation Matrix to Covariance Matrix Conversion*

**Description**

Function to convert a correlation matrix to a covariance matrix.

**Usage**

`cor2cov(cor.mat, sd)`

**Arguments**

- **cor.mat**: the correlation matrix to be converted
- **sd**: a vector that contains the standard deviations of the variables in the correlation matrix

**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).

**Author(s)**

Ken Kelley (Indiana University; ⟨KKIII@Indiana.Edu⟩), Keke Lai

---

**cv**  
*Function to calculate the regular (and 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 unbiased estimate for the standard deviation.

Author(s)

Ken Kelley (Indiana University; ⟨KKIII@Indiana.Edu⟩)

Examples

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

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", "bottom-left", "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 (Indiana University <laik@indiana.edu>), Ken Kelley (Indiana University <kkiii@indiana.edu>)

References

intr.plot

Regression Surface Containing Interaction

Description

To plot a three dimensional 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.

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" )
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 (Indiana University <laik@indiana.edu>), Ken Kelley (Indiana University <kkiii@indiana.edu>)

References

See Also

'`intr.plot.2d'`, 'persp'

Examples

```r
## 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)
```

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

```r
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
s.u

Unbiased estimate for the 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 unbiased 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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)
signal.to.noise.R2

References


Examples

```r
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)
```

**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

```r
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: \( \phi2.hat = R^2/(1-R^2) \)

phi2.adj.hat  Estimate of the signal-to-noise ratio using the usual adjusted R Square in place of R Square: \( \phi2.hat = \text{Adj.R2} / (1-\text{Adj.R2}) \)

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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References


See Also

'ci.R2', 'ss.aipe.R2'

Examples

```r
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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References


### See Also
'smd.c', 'conf.limits.nct', 'ss.aipe'

### Examples

```r
# Generate sample data.
s.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

```r
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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References


See Also

'smd', 'conf.limits.nct'

Examples

```r
# Generate sample data.
sd.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.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
certainty 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.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`) 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). 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 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 (In press) 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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References


See Also

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

Examples

# 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)
ss.aipe.R2.sensitivity

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 of at a given value of sample size
- **degree.of.certainty**: parameter to ensure confidence interval width with a specified degree of certainty
- **assurance**, **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 ...

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/independent variables), yet the function assumes random multivariate normal predictors (having a p+1 multivariate distribution). As Gatsonis and Sampson (1989) note in the context of statistical power analysis (recall this function is used in the context of precision), there is little difference in the outcome.
In the behavioral, educational, and social sciences, predictor variables are almost always random, and thus Random.Predictors should generally be used. Random.Predictors=TRUE specifies how both the sample size planning procedure and the confidence intervals are calculated based on the random predictors/regressors. The internal simulation generates random or fixed predictors/regressors based on whether variables predictor variables are random or fixed. However, when Random.Predictors=FALSE, only the sample size planning procedure and the confidence intervals are calculated based on the parameter. The parameter Generate.Random.Predictors (where the default is TRUE so that random predictors/regressors are generated) allows random or fixed predictor variables to be generated. Because the sample size planning procedure and the internal simulation are both specified, for purposes of sensitivity analysis random/fixed can be crossed to examine the effects of specifying sample size based on one but using it on data based on the other.

Value

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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References


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


See Also


Examples

# 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)
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 variance and not its square root (i.e., the standard deviation of the errors).

Author(s)
Ken Kelley (Indiana University; ⟨KKIII@Indiana.Edu⟩), Keke Lai

References
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 (Indiana University; ⟨KKIII@Indiana.Edu⟩), Keke Lai

References


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:

\[
\text{ss.aipe.c.ancova(error.var.anova=40, rho=.22, c.weights=c(1, -0.5, -0.5), width=3, assurance=.90)}
\]
Usage

ss.aipe.c.sensitivity.ancova(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 = 100, 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 \texttt{mu.y}, \texttt{mu.x}, \texttt{sigma.y}, and \texttt{sigma.x} are used to generate random data in the simulations for the sensitivity analysis. The value of \texttt{mu.y} should be the same as the square root of \texttt{true.error.var.anova}

So far this function is based on one-covariate randomized ANCOVA design only. The argument \texttt{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

\( \text{Psi obs} \) the observed (unstandardized) contrast
\( \text{se.Psi} \) the standard error of the observed (unstandardized) contrast
\( \text{se.Psi.restricted} \) the standard error of the observed (unstandardized) contrast calculated by ignoring the covariate
\( \text{se.res.over.se.full} \) the ratio of contrast’s full standard error over the restricted one in each iteration
\( \text{width.obs} \) full confidence interval width

Type I error:
- \( \text{Type I.Error} \) Type I error happens in each iteration
- \( \text{Type I.Error.Upper} \) Type I error happens in the upper end in each iteration
- \( \text{Type I.Error.Lower} \) Type I error happens in the lower end in each iteration

Type I error percentages:
- \( \text{Type I.Error} \) percentage of Type I error happened in the entire simulation
- \( \text{Type I.Error.Upper} \) percentage of Type I error happened in the upper end in the entire simulation
- \( \text{Type I.Error.Lower} \) percentage of Type I error happened in the lower end in the entire simulation

\( \text{width.NARROWER.than.desired} \) percentage of obtained widths that are narrower than the desired width

\( \text{Mean.width.obs} \) mean width of the obtained full confidence intervals
\( \text{Median.width.obs} \) median width of the obtained full confidence intervals
\( \text{Mean.se.res.vs.se.full} \) the mean of the ratios of contrast’s full standard error over the restricted one

\( \text{Psi.pop} \) population (unstandardized) contrast
\( \text{Contrast.Weights} \) 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

\( \text{Sample.Size.per.Group} \) sample size per group
\( \text{conf.level} \) the desired confidence interval coverage, (i.e., 1 - Type I error rate)
\( \text{assurance} \) specified assurance
\( \rho \) population correlation coefficient of the response and the covariate
\( \text{est.rho} \) estimated correlation coefficient of the response and the covariate

\( \text{true.error.var.ANOVA} \) population error variance of the ANOVA model
\( \text{est.error.var.ANOVA} \) estimated error variance of the ANOVA model
Author(s)

Keke Lai (Indiana University; ⟨LaiK@Indiana.Edu⟩)

Examples

```r
ss.aipe.c.sensitivity.ancova(true.error.var.ancova=30,
est.error.var.ancova=30, rho=.2, mu.y=c(10,12,15,13), mu.x=2,
G=10, sigma.x=1.3, sigma.y=2, c.weights=c(1,0,-1,0), width=3)

ss.aipe.c.sensitivity.ancova(true.error.var.anova=36,
est.error.var.anova=36, rho=.2, est.rho=.2, G=10,
mu.y=c(10,12,15,13), mu.x=2, sigma.x=1.3, sigma.y=6,
c.weights=c(1,0,-1,0), width=3, assurance=NULL)
```

---

**ss.aipe.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 wider than desired.

**Usage**

```r
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 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)
ss.aipe.cv.sensitivity

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 (Indiana University; ⟨KKIII@Indiana.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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)
ss.aipe.rc

See Also
cv.ss.aipe.cv

---

**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**

```r
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/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 (Indiana University; KKIII@Indiana.Edu)

**References**


**See Also**

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

**Examples**

```r
# 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.sensitivity

Sensitivity analysis for sample size planning from the Accuracy in Parameter Estimation Perspective for the unstandardized regression coefficient

Description
Performs a sensitivity analysis when planning sample size from the Accuracy in Parameter Estimation Perspective for the unstandardized regression coefficient.

Usage

Arguments
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>True.Var.Y</td>
<td>Population variance of the dependent variable (Y)</td>
</tr>
<tr>
<td>True.Cov.YX</td>
<td>Population covariances vector between the p predictor variables and the depen-</td>
</tr>
<tr>
<td></td>
<td>dent variable (Y)</td>
</tr>
<tr>
<td>True.Cov.XX</td>
<td>Population covariance matrix of the p predictor variables</td>
</tr>
<tr>
<td>Estimated.Var.Y</td>
<td>Estimated variance of the dependent variable (Y)</td>
</tr>
<tr>
<td>Estimated.Cov.YX</td>
<td>Estimated covariances vector between the p predictor variables and the depen-</td>
</tr>
<tr>
<td></td>
<td>dent variable (Y)</td>
</tr>
<tr>
<td>Estimated.Cov.XX</td>
<td>Estimated Population covariance matrix of the p predictor variables</td>
</tr>
<tr>
<td>Specified.N</td>
<td>Directly specified sample size (instead of using Estimated.Rho.YX and Esti-</td>
</tr>
<tr>
<td></td>
<td>mated.RHO.XX)</td>
</tr>
<tr>
<td>which.predictor</td>
<td>identifies which of the p predictors is of interest</td>
</tr>
<tr>
<td>w</td>
<td>desired confidence interval width for the regression coefficient of interest</td>
</tr>
<tr>
<td>Noncentral</td>
<td>specify with a TRUE/FALSE statement whether or not the noncentral approach</td>
</tr>
<tr>
<td></td>
<td>to sample size planning should be used</td>
</tr>
<tr>
<td>Standardize</td>
<td>specify with a TRUE/FALSE statement whether or not the regression coefficientwill be standardized; default is TRUE</td>
</tr>
</tbody>
</table>
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 (Indiana University; KKIII@Indiana.Edu)

References


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 jth predictor variable from the remaining p-1 predictor variables
p  the number of predictor variables
b.j  the regression coefficient for the jth 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 jth 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/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
cnf.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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References

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

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.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.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


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/FALSE statement whether or not the noncentral approach to sample size planning should be used
**ss.aipe.reg.coef.sensitivity**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standardize</td>
<td>specify with a TRUE/FALSE statement whether or not the regression coefficient will be standardized</td>
</tr>
<tr>
<td>conf.level</td>
<td>desired level of confidence for the computed interval (i.e., 1 - the Type I error rate)</td>
</tr>
<tr>
<td>degree.of.certainty</td>
<td>degree of certainty that the obtained confidence interval will be sufficiently narrow</td>
</tr>
<tr>
<td>assurance</td>
<td>an alias for degree.of.certainty</td>
</tr>
<tr>
<td>certainty</td>
<td>an alias for degree.of.certainty</td>
</tr>
<tr>
<td>G</td>
<td>the number of generations/replication of the simulation student within the function</td>
</tr>
<tr>
<td>print.iter</td>
<td>specify with a TRUE/FALSE statement if the iteration number should be printed as the simulation within the function runs</td>
</tr>
</tbody>
</table>

**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 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.

**Author(s)**

Ken Kelley (Indiana University; ⟨KKIII@Indiana.Edu⟩)

**References**


**See Also**

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

```r
ss.aipe.sc(psi, c.weights, width, conf.level = 0.95, assurance = NULL, certainty = ...)
```

**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 (Indiana University; ⟨KKIII@Indiana.Edu⟩), Keke Lai

**References**


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 group such that the width
# of 95 percent confidence interval of the standardized
# contrast is, with 90 percent assurance, no wider than .4:
ss.aipe.sc(psi=.6, c.weights=c(.5, .5, -.5, -.5, 0), width=.4, assurance=.90)
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 simu-
lation 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 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

Author(s)
Ken Kelley (Indiana University; ⟨KKIII@Indiana.Edu⟩), Keke Lai

References


See Also
ss.aipe.sc, ss.aipe.c, conf.limits.nct

Examples
ss.aipe.sc.sensitivity(true.psi=.5, estimated.psi = .5, c.weights=c(1,-1), desired.width = .4, assurance = .95, G=10)
ss.aipe.sm

ss.aipe.sc.sensitivity(true.psi=.5, selected.n=203, c.weights=c(1,-1),
desired.width =.4, assurance =.95, G=10)

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 (Indiana University; ⟨KKIII@Indiana.Edu⟩), Keke Lai

References

intervals that are based on central and noncentral distributions, Educational and Psychological Mea-
surement, 61, 532–574.


Kelley, K. (2005) The effects of nonnormal distributions on confidence intervals around the stan-
dardized mean difference: Bootstrap and parametric confidence intervals, Educational and Psychol-


**See Also**

conf.limit.nct, ci.sm

**Examples**

```r
# 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
# 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**

```r
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
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>selected.n</td>
<td>selected sample size to use in order to determine distributional properties of at a given value of sample size</td>
</tr>
<tr>
<td>assurance</td>
<td>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)</td>
</tr>
<tr>
<td>certainty</td>
<td>an alias for assurance</td>
</tr>
<tr>
<td>conf.level</td>
<td>the desired confidence interval coverage, (i.e., 1 - Type I error rate)</td>
</tr>
<tr>
<td>G</td>
<td>number of generations (i.e., replications) of the simulation</td>
</tr>
<tr>
<td>print.iter</td>
<td>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</td>
</tr>
<tr>
<td>...</td>
<td>allows one to potentially include parameter values for inner functions</td>
</tr>
</tbody>
</table>

**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 (Indiana University; ⟨KKIII@Indiana.Edu⟩), Keke Lai

References
intervals that are based on central and noncentral distributions, Educational and Psychological Mea-
surement, 61, 532–574.


Kelley, K. (2005) The effects of nonnormal distributions on confidence intervals around the stan-
dardized mean difference: Bootstrap and parametric confidence intervals, Educational and Psycho-

Kelley, K. (In press). Constructing confidence intervals for standardized effect sizes: Theory, appli-
cation, and implementation. Journal of Statistical Software.

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 statistic-
ical 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.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)
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 upper confidence limits is approximate, but very close to being exact. The pt() function is limited to accurate values when 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 (Indiana University; KKIII@Indiana.Edu)

References


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

```r
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)
```

Description

Performs sensitivity analysis for sample size determination for the standardized mean difference given a population and an 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

```r
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 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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References


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 
kth predictor variable from the remaining p-1 predictor variables
K the number of predictor variables
beta.k the regression coefficient for the kth 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 (in press), 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.

0.1 Warning

As discussed in Kelley and Maxwell (in press), the sample size planning approach from the AIPE perspective used in this function is only an approximation.

Author(s)

Ken Kelley (Indiana University; KKIII@Indiana.Edu)

References


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)

Description

Performs a sensitivity analysis when planning sample size from the Accuracy in Parameter Estimation Perspective for the standardized regression coefficient.

Usage

```r
```

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/FALSE statement whether or not the noncentral approach to sample size planning should be used
- `Standardize`: specify with a TRUE/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 (Indiana University; KKIII@Indiana.Edu)

References

See Also
'ss.aipe.reg.coef.sensitivity', 'ss.aipe.rc.sensitivity', 'ss.aipe.reg.coef', 'ci.reg.coef'
### 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

```r
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)
- **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
- ... (not currently implemented) 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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

See Also

- `ss.aipe.R2`
- `ss.power.reg.coef`
- `conf.limits.ncf`

Examples

```r
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

```r
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:

\[
\]

**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

Determine 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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References

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.YX.Maxwell <- Cor.Mat.Maxwell[1,2:6]
R2.Maxwell <- Rho.YX.Maxwell

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 because 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**

```r
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:

\[
\]

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

Determine 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.
ss.power.reg.coef

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 (Indiana University; ⟨KKIII@Indiana.Edu⟩)

References


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.YX.Maxwell <- Cor.Mat.Maxwell[1,2:6]
R2.Maxwell <- Rho.YX.Maxwell

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 because 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)
Description

Functions useful for converting a standardized mean difference to a noncentrality parameter, and vice versa.

Usage

\[
\text{lambda2delta}(\text{lambda}, \text{n.1}, \text{n.2}) \\
\text{delta2lambda}(\text{delta}, \text{n.1}, \text{n.2})
\]

Arguments

- \text{lambda} noncentral value from a \text{t}-distribution
- \text{delta} population value of the standardized mean difference
- \text{n.1} sample size in group 1
- \text{n.2} sample size in group 2

Details

Although \text{lambda} is the population noncentral value, it can be regarded as the observed value of a \text{t}-statistics. Likewise, \text{delta} can be regarded as the observed standardized mean difference. Thus, the observed standardized mean difference can be converted to the observed \text{t}-value. These functions are especially helpful in the context of forming confidence intervals around the population standardized mean difference.

Value

Either the value of \text{delta} given \text{lambda} or \text{lambda} given \text{delta} (and the per-group sample sizes).

Author(s)

Ken Kelley (Indiana University; ⟨KKIII@Indiana.Edu⟩)

See Also

'smd', 'ci.smd', 'ss.aipe.smd'

Examples

\[
\text{lambda2delta}(\text{lambda}=2, \text{n.1}=113, \text{n.2}=113) \\
\text{delta2lambda}(\text{delta}=.266076, \text{n.1}=113, \text{n.2}=113)
\]
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
spedify 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 verfiy.ss=TRUE in the ss.aipe.R2 function. Although users can use verify.ss.aipe.R2 directly, it is not recomended.

Value

Returns the exact (provided G is large enough) sample size necessary to satisfy the conditions specified.
Author(s)

Ken Kelley (Indiana University; ⟨KKIII@Indiana.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 (Indiana University; ⟨KKIII@indiana.edu⟩) and Po-Ju Wu (Indiana University; ⟨pojwu@indiana.edu⟩)

See Also
par, nlme, vit.fitted,

Examples

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)
**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**

```r
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 use 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 (Indiana University; ⟨KKIII@indiana.edu⟩) and Po-Ju Wu (Indiana University; ⟨pojwu@indiana.edu⟩)
See Also

par, nlme, lme4, lme, lmer, vit.fitted

Examples

data(Gardner.LD)
library(nlme)
grouped.Gardner.LD <- groupedData(Score ~ Trial|ID, data=Gardner.LD, order.groups=FALSE)
G.L.nlsList <- nlsList(SSlogis,grouped.Gardner.LD)
G.L.nlme <- nlme(G.L.nlsList)

vit.fitted(G.L.nlme)
vit.fitted(G.L.nlme, pct.rand = 50)
Index

*Topic datasets
  Gardner.LD, 4
  HS.data, 5
  prof.salary, 52

*Topic design
  aipe.smd, 9
  ci.c, 15
  ci.c.ancova, 17
  ci.pvaf, 20
  ci.R, 11
  ci.rc, 21
  ci.sc, 25
  ci.sc.ancova, 27
  ci.sm, 28
  ci.snr, 33
  ci.src, 35
  ci.srsnr, 37
  conf.limits.nc.chisq, 38
  conf.limits.ncf, 40
  cor2cov, 47
  cv, 47
  Expected.R2, 1
  s.u, 53
  ss.aipe.c, 64
  ss.aipe.c.ancova, 65
  ss.aipe.c.sensitivity.ancova, 66
  ss.aipe.cv, 69
  ss.aipe.cv.sensitivity, 70
  ss.aipe.R2, 59
  ss.aipe.R2.sensitivity, 61
  ss.aipe.rc, 72
  ss.aipe.rc.sensitivity, 74
  ss.aipe.reg.coef, 76
  ss.aipe.reg.coef.sensitivity, 78
  ss.aipe.sc, 80

  ss.aipe.sc.sensitivity, 81
  ss.aipe.sm, 84
  ss.aipe.sm.sensitivity, 85
  ss.aipe.smd, 88
  ss.aipe.smd.sensitivity, 90
  ss.aipe.src, 93
  ss.aipe.src.sensitivity, 96
  ss.aipe.src.sensitivity, 98
  ss.aipe.rc, 99
  ss.aipe.rc.sensitivity, 100
  t.and.smd.conversion, 106
  Variance.R2, 8
  verify.ss.aipe.R2, 107

*Topic device
  vit, 108
  vit.fitted, 110

*Topic dynamic
  vit, 108
  vit.fitted, 110

*Topic hplot
  vit, 108
  vit.fitted, 110

*Topic htest
  aipe.smd, 9
  ci.cv, 18
  ci.R2, 12
  ci.reg.coef, 23
  ci.smd, 30
  ci.smd.c, 32
  conf.limits.nct, 45
  conf.limits.nct.M1, 41
  conf.limits.nct.M2, 42
  conf.limits.nct.M3, 43
  cv, 47
  s.u, 53
  signal.to.noise.R2, 54
  smd, 55
  smd.c, 57
  ss.aipe.cv, 69
ss.aipe.cv.sensitivity, 70
ss.aipe.sm.sensitivity, 85
ss.aipe.smd, 88
ss.aipe.smd.sensitivity, 90

* Topic models
  ci.cv, 18
  ci.smd.c, 32
  conf.limits.nct, 45
  conf.limits.nct.M1, 41
  conf.limits.nct.M2, 42
  conf.limits.nct.M3, 43
  signal.to.noise.R2, 54
  smd, 55
  smd.c, 57

* Topic multivariate
  ci.R2, 12
  conf.limits.nc.chisq, 38
  conf.limits.ncf, 40

* Topic package
  MBESS, 7

* Topic regression
  ci.R, 11
  ci.R2, 12
  conf.limits.nc.chisq, 38
  conf.limits.ncf, 40
  intr.plot, 50
  intr.plot.2d, 48

* Topic univar
  ci.smd, 30

aipe.smd, 9

ci.c, 15
ci.c.ancova, 17
ci.cv, 18
ci.pvaf, 20
ci.R, 11
ci.R2, 12
ci.rc, 21
ci.reg.coef, 23
ci.sc, 25
ci.sc.ancova, 27
ci.sm, 28
ci.smd, 30
ci.smd.c, 32
ci.snr, 33
ci.src, 35

ci.srsnr, 37
conf.limits.nc.chisq, 38
conf.limits.ncf, 40
conf.limits.nct, 45
conf.limits.nct.M1, 41
conf.limits.nct.M2, 42
conf.limits.nct.M3, 43
cor2cov, 47
cv, 47
delta2lambda
  (t.and.smd.conversion), 106

Expected.R2, 1
F2Rsquare

Gardner.LD, 4
HS.data, 5
intr.plot, 50
intr.plot.2d, 48
lambda2delta
  (t.and.smd.conversion), 106
Lambda2Rsquare

MBES (MBESS), 7
mbes (MBESS), 7
MBESS, 7
mbess (MBESS), 7

prof.salary, 52
Rsquare2F
Rsquare2Lambda

s.u, 53
signal.to.noise.R2, 54
smd, 55
INDEX

smd.c, 57
ss.aipe.c, 64
ss.aipe.c.ancova, 65
ss.aipe.c.sensitivity.ancova, 66
ss.aipe.cv, 69
ss.aipe.cv.sensitivity, 70
ss.aipe.R2, 59
ss.aipe.R2.sensitivity, 61
ss.aipe.rc, 72
ss.aipe.rc.sensitivity, 74
ss.aipe.reg.coef, 76
ss.aipe.reg.coef.sensitivity, 78
ss.aipe.sc, 80
ss.aipe.sc.sensitivity, 81
ss.aipe.sm, 84
ss.aipe.sm.sensitivity, 85
ss.aipe.smd, 88
ss.aipe.smd.full (aipe.smd), 9
ss.aipe.smd.lower (aipe.smd), 9
ss.aipe.smd.sensitivity, 90
ss.aipe.smd.upper (aipe.smd), 9
ss.aipe.src, 93
ss.aipe.src.sensitivity, 96
ss.power.R2, 98
ss.power.rc, 99
ss.power.reg.coef, 102

t.and.smd.conversion, 106

Variance.R2, 8
verify.ss.aipe.R2, 107
vit, 108
vit.fitted, 110