

Package ‘metansue’

September 1, 2016

Type Package

Title Meta-Analysis of Studies with Non Statistically-Significant Unreported Effects

Version 1.2

Date 2016-08-31

Author Joaquim Radua

Maintainer Joaquim Radua <jradua@fidmag.com>

Description A novel meta-analytic method that allows an unbiased inclusion of studies with Non Statistically-Significant Unreported Effects (NSUEs). Briefly, the method first calculates the interval where the unreported effects (e.g. t-values) should be according to the threshold of statistical significance used in each study. Afterwards, maximizing likelihood techniques are used to impute the expected effect size of each study with NSUEs, accounting for between-study heterogeneity and potential covariates. Multiple imputations of the NSUEs are then randomly created based on the expected value, variance and statistical significance bounds. Finally, a restricted-maximum likelihood random-effects meta-analysis is separately conducted for each set of imputations, and estimations from these meta-analyses are pooled. Please read the reference in 'meta.nsue' for details of the procedure.

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2016-09-01 13:20:00

R topics documented:

forest	2
forest.meta.nsue	3
funnel	4
funnel.meta.nsue	5
leave1out	6
leave1out.nsue	7
linearHypothesis	8
linearHypothesis.meta.nsue	9

meta	10
meta.nsue	11
metabias	13
metabias.meta.nsue	14
metalm	15
metalm.meta.nsue	16
smc_from_t	17

Index	20
--------------	-----------

forest	<i>Forest Plots</i>
--------	---------------------

Description

Draws a forest plot.

Usage

```
forest(x, ...)
```

Arguments

x	an object of class "meta.nsue", or of other classes in other meta-analytic packages (e.g. "meta" or "rma").
...	other arguments.

Details

The function `forest` is generic. See [forest.meta.nsue](#) for plotting forest plots for "meta.nsue" objects, or other methods from other meta-analytic packages (e.g. [forest.meta](#) or [forest.rma](#)).

See Also

[forest.meta.nsue](#) for plotting forest plots for "meta.nsue" objects.

[funnel](#) for plotting funnel plots.

[meta](#) for conducting a meta-analysis.

[linearHypothesis](#) for changing the hypotheses.

[pdf](#) for saving the plot in a PDF file.

forest.meta.nsue *Forest Plots for “meta.nsue” Objects*

Description

Draws a forest plot.

Usage

```
## S3 method for class 'meta.nsue'  
forest(x, ...)
```

Arguments

x an object of class "meta.nsue".
... other arguments (currently ignored).

Details

The plot shows the observed or imputed effect sizes of the studies with the corresponding 95% confidence intervals. A light gray shadow shows the interval containing 95% of the imputations for studies with Non Statistically-Significant Unreported Effects (NSUEs).

The effect sizes of the hypotheses of the model with the corresponding 95% confidence intervals are added to the bottom of the forest plot. Note that in case of complex models you may want to show only those hypotheses related to the overall or subgroup means, and thus you may have to specify them using [linearHypothesis](#).

Value

The function invisibly returns the optimal width and height of the plot, useful when calling the function [pdf](#).

Author(s)

Joaquim Radua

See Also

[funnel](#) for plotting funnel plots.
[meta](#) for conducting a meta-analysis.
[linearHypothesis](#) for changing the hypotheses.
[pdf](#) for saving the plot in a PDF file.

Examples

```
t <- c(3.4, NA, NA, NA, NA, 2.8, 2.1, 3.1, 2.0, 3.4)
n <- c(40, 20, 22, 24, 18, 30, 25, 30, 16, 22)
m <- meta(smc_from_t(t, n))
forest(m)
```

funnel

Funnel Plots

Description

Draws a funnel plot.

Usage

```
funnel(x, ...)
```

Arguments

x	an object of class "meta.nsue", or of other classes in other meta-analytic packages (e.g. "meta" or "rma").
...	other arguments.

Details

The function `funnel` is generic. See [funnel.meta.nsue](#) for plotting funnel plots for "meta.nsue" objects, or other methods from other meta-analytic packages (e.g. [funnel.meta](#) or [funnel.rma](#)).

See Also

[funnel.meta.nsue](#) for plotting funnel plots for "meta.nsue" objects.

[metabias](#) for testing for funnel plot asymmetry.

[forest](#) for plotting forest plots.

[meta](#) for conducting a meta-analysis.

[pdf](#) for saving the plot in a PDF file.

`funnel.meta.nsue`*Funnel Plots for “meta.nsue” Objects*

Description

Draws a funnel plot.

Usage

```
## S3 method for class 'meta.nsue'  
funnel(x, ...)
```

Arguments

`x` an object of class “meta.nsue”.
`...` other arguments (currently ignored).

Details

The plot shows the residual effect size of the studies on the x-axis against their standard errors on the y-axis. A light gray shadow shows the ellipse approximately containing 95% of the imputations for studies with Non Statistically-Significant Unreported Effects (NSUEs). A pseudo confidence interval region is drawn with a dashed line.

Asymmetry in the funnel plot may be an indication of publication bias, but this should be taken with caution, especially if sample sizes are too similar, if there are subgroups, or if dichotomous outcomes have been used (some effect estimates are naturally correlated with their standard errors).

Also, publication bias cannot be excluded in case of negative results, whereas it is only one of the possible explanations in case of positive results - others being selective outcome reporting, poor methodological quality in the smaller studies, true heterogeneity (i.e. effect size truly depends on study size), and etcetera.

Author(s)

Joaquim Radua

References

Egger, M., Smith, G.D., Schneider, M., Minder, C. (1997) Bias in meta-analysis detected by a simple, graphical test. *BMJ*, **315**, 629–634, doi:10.1136/bmj.315.7109.629.

See Also

[metabias](#) for testing for funnel plot asymmetry.

[forest](#) for plotting forest plots.

[meta](#) for conducting a meta-analysis.

[pdf](#) for saving the plot in a PDF file.

Examples

```
t <- c(3.4, NA, NA, NA, NA, 2.8, 2.1, 3.1, 2.0, 3.4)
n <- c(40, 20, 22, 24, 18, 30, 25, 30, 16, 22)
m <- meta(smc_from_t(t, n))
funnel(m)
```

leave1out

Compute Leave-One-Out Diagnostics

Description

Repeatedly fits the specified model, leaving out one study at a time.

Usage

```
leave1out(x, ...)
```

Arguments

`x` an object of class "nsue", or of other classes in other meta-analytic packages (e.g. "rma").

`...` other arguments.

Details

The function `leave1out` is generic. See [leave1out.nsue](#) for computing leave-one-out diagnostics for "nsue" objects, or other methods from other meta-analytic packages (e.g. [leave1out.rma.uni](#)).

See Also

[leave1out.nsue](#) for computing leave-one-out diagnostics for "nsue" objects.

[smc_from_t](#), [smd_from_t](#), [z_from_r](#) and [r_in_smd_from_t_means_and_sds1](#) for creating "nsue" objects.

[meta](#) for conducting a meta-analysis.

Description

Repeatedly fits the specified model, leaving out one study at a time.

Usage

```
## S3 method for class 'nsue'
leave1out(x, data = data.frame(), formula = ~1, hypotheses = NULL,
          n.imp = 50, n.bins = 200, maxiter = 200, tol = 1e-06, ...)
```

Arguments

x	an object of class "nsue".
data	an optional data frame containing variables to be used as regressors in the maximum likelihood step.
formula	an object of class "formula": a symbolic description of the model to be fitted.
hypotheses	a list of hypotheses, or NULL to simply test the coefficients of the model.
n.imp	number of imputations of NSUEs per study.
n.bins	number of bins used in the imputations.
maxiter	maximum number of iterations in the REML estimation of τ^2 .
tol	tolerance in the REML estimation of τ^2 .
...	other arguments (currently ignored).

Details

Use [smc_from_t](#), [smd_from_t](#), [z_from_r](#) or [r_in_smd_from_t_means_and_sds1](#) to create "nsue" objects.

Models for [meta.nsue](#), [leave1out.nsue](#) and [metalm.meta.nsue](#) are specified symbolically. The formula is a series of terms which specify a linear predictor for x. A formula specification of the form `first + second` indicates a multiple regression by `first` and `second`. A specification of the form `first:second` indicates the interaction of `first` with `second`. The specification `first*second` is the same as `first + second + first:second`.

Each hypothesis must be a matrix (or vector) giving linear combinations of coefficients by rows.

Value

`leave1out.nsue` returns an object of class "leave1out.nsue", which is a list containing a list for each iteration with the component `study` (the study discarded) and the component `meta.nsue`, which is an object of class "meta.nsue".

The functions `print` and `summary` may be used to print the details or a summary of the results.

Author(s)

Joaquim Radua

References

Radua, J., Schmidt, A., Borgwardt, S., Heinz, A., Schlagenhauf, F., McGuire, P., Fusar-Poli, P. (2015) Ventral striatal activation during reward processing in psychosis. A neurofunctional meta-analysis. *JAMA Psychiatry*, **72**, 1243–51.

See Also

[smc_from_t](#), [smd_from_t](#), [z_from_r](#) and [r_in_smd_from_t_means_and_sds1](#) for creating "nsue" objects.

[meta](#) for conducting a meta-analysis.

Examples

```
t <- c(3.4, NA, NA, NA, NA, 2.8, 2.1, 3.1, 2.0, 3.4)
n <- c(40, 20, 22, 24, 18, 30, 25, 30, 16, 22)
leave1out(smc_from_t(t, n))
```

linearHypothesis	<i>Test Linear Hypothesis</i>
------------------	-------------------------------

Description

Function for testing a linear hypothesis. Along with [metalm](#) (or others depending on the class, e.g. "lm"), it can be used to carry out meta-regression, meta-comparisons, analyses of variance and covariance, and etcetera.

Usage

```
linearHypothesis(x, ...)
```

Arguments

x	an object of class "meta.nsue", or of other classes in other packages (e.g. "glm", "lm", "lme", "mer", "merMod", "mlm", "nlsList", "polr" or "svyglm").
...	other arguments.

Details

The function `linearHypothesis` is generic. See [linearHypothesis.meta.nsue](#) for testing linear hypotheses for "meta.nsue" objects, or other methods from other packages (e.g. [linearHypothesis.default](#)).

See Also

[linearHypothesis.meta.nsue](#) for testing linear hypotheses for "meta.nsue" objects.

[metalm](#) for fitting meta-analytic models.

[meta](#) for conducting a meta-analysis.

linearHypothesis.meta.nsue

Test Linear Hypothesis for "meta.nsue" Objects

Description

Function for testing a linear hypothesis. Along with [metalm](#), it can be used to carry out meta-regression, meta-comparisons, analyses of variance and covariance, and etcetera.

Usage

```
## S3 method for class 'meta.nsue'  
linearHypothesis(x, hypotheses, ...)
```

Arguments

x	an object of class "meta.nsue".
hypotheses	a list of hypotheses, or NULL to test the coefficients of the model.
...	other arguments (currently ignored).

Details

Each hypothesis must be a matrix (or vector) giving linear combinations of coefficients by rows.

Value

`linearHypothesis.meta.nsue` returns an object of class "meta.nsue".

The functions `print` and `summary` may be used to print the details or a summary of the results. The generic accessor functions `coefficients`, `fitted.values` and `residuals` extract various useful features of the value returned by `linearHypothesis.meta.nsue`.

Author(s)

Joaquim Radua

References

Radua, J., Schmidt, A., Borgwardt, S., Heinz, A., Schlagenhauf, F., McGuire, P., Fusar-Poli, P. (2015) Ventral striatal activation during reward processing in psychosis. A neurofunctional meta-analysis. *JAMA Psychiatry*, **72**, 1243–51.

See Also

[metalm](#) for fitting meta-analytic models.

[meta](#) for conducting a meta-analysis.

Examples

```
t <- c(3.4, NA, NA, NA, NA, 2.8, 2.1, 3.1, 2.0, 3.4)
n <- c(40, 20, 22, 24, 18, 30, 25, 30, 16, 22)
m <- meta(smc_from_t(t, n))
mean.age <- rnorm(10, 30, 5)
m <- metalm(m, ~ mean.age)
linearHypothesis(m, list(age_10years = c(0, 10)))
```

meta

Meta-Analysis

Description

Conduct a meta-analysis.

Usage

```
meta(x, ...)
```

Arguments

x	an object of class "nsue", or of other classes in other meta-analytic packages.
...	other arguments.

Details

The function meta is generic. See [meta.nsue](#) for conducting a meta-analysis for "nsue" objects.

See Also

[meta.nsue](#) for conducting a meta-analysis for "nsue" objects.

[smc_from_t](#), [smd_from_t](#), [z_from_r](#) and [r_in_smd_from_t_means_and_sds1](#) for creating "nsue" objects.

[forest](#) for plotting forest plots.

[funnel](#) for plotting funnel plots.

[metabias](#) for testing for funnel plot asymmetry.

[leave1out](#) for computing leave-one-out diagnostics.

[metalm](#) for fitting meta-analytic models.

[linearHypothesis](#) for testing linear hypotheses.

meta.nsue	<i>Meta-Analysis of Studies with Non Statistically-Significant Unreported Effects</i>
-----------	---

Description

Conduct a meta-analysis. MetaNSUE is a meta-analytic method that allows an unbiased inclusion of studies with Non Statistically-Significant Unreported Effects (NSUEs).

Usage

```
## S3 method for class 'nsue'
meta(x, data = data.frame(), formula = ~1, hypotheses = NULL,
      n.imp = 50, n.bins = 200, maxiter = 200, tol = 1e-06, ...)
```

Arguments

x	an object of class "nsue".
data	an optional data frame containing variables to be used as regressors in the maximum likelihood step.
formula	an object of class "formula": a symbolic description of the model to be fitted.
hypotheses	a list of hypotheses, or NULL to test the coefficients of the model.
n.imp	number of imputations of NSUEs.
n.bins	number of bins used in the imputations.
maxiter	maximum number of iterations in the REML estimation of τ^2 .
tol	tolerance in the REML estimation of τ^2 .
...	other arguments (currently ignored).

Details

Use `smc_from_t`, `smd_from_t`, `z_from_r` or `r_in_smd_from_t_means_and_sds1` to create "nsue" objects.

Models for `meta.nsue`, `leave1out.nsue` and `metalm.meta.nsue` are specified symbolically. The formula is a series of terms which specify a linear predictor for x . A formula specification of the form `first + second` indicates a multiple regression by `first` and `second`. A specification of the form `first:second` indicates the interaction of `first` with `second`. The specification `first*second` is the same as `first + second + first:second`.

Each hypothesis must be a matrix (or vector) giving linear combinations of coefficients by rows.

Value

meta.nsue returns an object of class "meta.nsue", which is a list containing the following components:

measure	the effect-size measure used.
known	a list with the known effect sizes and their indexes.
unknown	a list with the imputations of NSUEs and their indexes.
y.var	the variances of the effect sizes.
y2var_k1	a constant needed to derive the variances.
y2var_k2	a constant needed to derive the variances.
labels	the labels of the studies.
rm	a list with the expected correlation between repeated-measures studies, a conversion matrix and the study weights.
heterogeneity	a list with τ^2 , H^2 , I^2 and Q test.
model	a list with the formula, matrix and coefficients of the model.
hypotheses	a list with the matrixes and coefficients of the hypotheses.

The functions `print` and `summary` may be used to print the details or a summary of the results. The function `subset` returns the subset of studies that meets a condition. The generic accessor functions `coefficients`, `fitted.values` and `residuals` extract various useful features of the value returned by `meta.nsue`.

Author(s)

Joaquim Radua

References

Radua, J., Schmidt, A., Borgwardt, S., Heinz, A., Schlagenhauf, F., McGuire, P., Fusar-Poli, P. (2015) Ventral striatal activation during reward processing in psychosis. A neurofunctional meta-analysis. *JAMA Psychiatry*, **72**, 1243–51.

See Also

[smc_from_t](#), [smd_from_t](#), [z_from_r](#) and [r_in_smd_from_t_means_and_sds1](#) for creating "nsue" objects.

[forest](#) for plotting forest plots.

[funnel](#) for plotting funnel plots.

[metabias](#) for testing for funnel plot asymmetry.

[leave1out](#) for computing leave-one-out diagnostics.

[metalm](#) for fitting meta-analytic models.

[linearHypothesis](#) for testing linear hypotheses.

Examples

```
t <- c(3.4, NA, NA, NA, NA, 2.8, 2.1, 3.1, 2.0, 3.4)
n <- c(40, 20, 22, 24, 18, 30, 25, 30, 16, 22)
meta(smc_from_t(t, n))
```

metabias

Test for Funnel Plot Asymmetry

Description

Test for funnel plot asymmetry.

Usage

```
metabias(x, ...)
```

Arguments

x	an object of class "meta.nsue", or of other classes in other meta-analytic packages (e.g. "meta").
...	other arguments.

Details

The function `metabias` is generic. See [metabias.meta.nsue](#) for testing for funnel plot asymmetry for "meta.nsue" objects, or other methods from other meta-analytic packages (e.g. [metabias.meta](#)).

See Also

[metabias.meta.nsue](#) for testing for funnel plot asymmetry for "meta.nsue" objects.

[funnel](#) for plotting funnel plots.

[meta](#) for conducting a meta-analysis.

[metalm](#) for fitting meta-analytic models.

metabias.meta.nsue *Test for Funnel Plot Asymmetry for “meta.nsue” Objects*

Description

Test for funnel plot asymmetry.

Usage

```
## S3 method for class 'meta.nsue'
metabias(x, maxiter = 100, tol = 1e-06, ...)
```

Arguments

x	an object of class "meta.nsue".
maxiter	maximum number of iterations in the REML estimation of τ^2 .
tol	tolerance in the REML estimation of τ^2 .
...	other arguments (currently ignored).

Details

The test, based on a meta-regression by the standard error, aims to detect asymmetry in the funnel plot, which may be an indication of publication bias.

However, results should be taken with caution, especially if there are too few studies (at least 10 studies were suggested by Sterne et al. (2011)), if sample sizes are too similar, if there are outliers or influential studies or subgroups in the plot, or if dichotomous outcomes have been used (some effect estimates are naturally correlated with their standard errors).

Also, publication bias cannot be excluded in case of negative results, whereas it is only one of the possible explanations in case of positive results - others being selective outcome reporting, poor methodological quality in the smaller studies, true heterogeneity (i.e. effect size truly depends on study size), and etcetera (Egger 1997).

Value

A list with class "htest" containing the following components:

data.name	a character string giving the name of the data.
method	a character string indicating that a "meta.nsue" regression test for funnel plot asymmetry was performed.
statistic	the value of the z-statistic.
p.value	the p-value for the test.

Author(s)

Joaquim Radua

References

- Egger, M., Smith, G.D., Schneider, M., Minder, C. (1997) Bias in meta-analysis detected by a simple, graphical test. *BMJ*, **315**, 629–634, doi:10.1136/bmj.315.7109.629.
- Radua, J., Schmidt, A., Borgwardt, S., Heinz, A., Schlagenhauf, F., McGuire, P., Fusar-Poli, P. (2015) Ventral striatal activation during reward processing in psychosis. A neurofunctional meta-analysis. *JAMA Psychiatry*, **72**, 1243–51.
- Sterne, J.A., Sutton, A.J., Ioannidis, J.P., Terrin, N., Jones, D.R., Lau, J., Carpenter, J., Rucker, G., Harbord, R.M., Schmid, C.H., Tetzlaff, J., Deeks, J.J., Peters, J., Macaskill, P., Schwarzer, G., Duval, S., Altman, D.G., Moher, D., Higgins, J.P. (2011) Recommendations for examining and interpreting funnel plot asymmetry in meta-analyses of randomised controlled trials. *BMJ*, **343**, d4002, doi:10.1136/bmj.d4002.

See Also

- [funnel](#) for plotting funnel plots.
- [meta](#) for conducting a meta-analysis.
- [metalm](#) for fitting meta-analytic models.

Examples

```
t <- c(3.4, NA, NA, NA, NA, 2.8, 2.1, 3.1, 2.0, 3.4)
n <- c(40, 20, 22, 24, 18, 30, 25, 30, 16, 22)
m <- meta(smc_from_t(t, n))
metabias(m)
```

metalm

Fitting Meta-Analytic Linear Models

Description

Fits meta-analytic linear models. Along with [linearHypothesis](#), it can be used to carry out meta-regression, meta-comparisons, analyses of variance and covariance, and etcetera.

Usage

```
metalm(x, ...)
```

Arguments

- `x` an object of class "meta.nsue", or of other classes in other meta-analytic packages.
- `...` other arguments.

Details

The function `metalm` is generic. See [metalm.meta.nsue](#) for fitting meta-analytic models for "meta.nsue" objects.

See Also

[metalm.meta.nsue](#) for fitting meta-analytic models for "meta.nsue" objects.

[meta](#) for conducting a meta-analysis.

[linearHypothesis](#) for testing linear hypotheses.

[metabias](#) for testing for funnel plot asymmetry.

 metalm.meta.nsue

Fitting Meta-Analytic Linear Models for "meta.nsue" Objects

Description

Fits meta-analytic linear models. Along with [linearHypothesis](#), it can be used to carry out meta-regression, meta-comparisons, analyses of variance and covariance, and etcetera.

Usage

```
## S3 method for class 'meta.nsue'
metalm(x, formula, maxiter = 100, tol = 1e-06, ...)
```

Arguments

x	an object of class "meta.nsue".
formula	an object of class "formula": a symbolic description of the model to be fitted.
maxiter	maximum number of iterations in the REML estimation of τ^2 .
tol	tolerance in the REML estimation of τ^2 .
...	other arguments (currently ignored).

Details

Models for [meta.nsue](#), [leave1out.nsue](#) and `metalm.meta.nsue` are specified symbolically. The formula is a series of terms which specify a linear predictor for x . A formula specification of the form `first + second` indicates a multiple regression by `first` and `second`. A specification of the form `first:second` indicates the interaction of `first` with `second`. The specification `first*second` is the same as `first + second + first:second`.

Value

`metalm.meta.nsue` returns an object of class "meta.nsue".

The functions `print` and `summary` may be used to print the details or a summary of the results. The generic accessor functions `coefficients`, `fitted.values` and `residuals` extract various useful features of the value returned by `metalm.meta.nsue`.

Author(s)

Joaquim Radua

References

Radua, J., Schmidt, A., Borgwardt, S., Heinz, A., Schlagenhauf, F., McGuire, P., Fusar-Poli, P. (2015) Ventral striatal activation during reward processing in psychosis. A neurofunctional meta-analysis. *JAMA Psychiatry*, **72**, 1243–1251.

See Also

[meta](#) for conducting a meta-analysis.

[linearHypothesis](#) for testing linear hypotheses.

[metabias](#) for testing for funnel plot asymmetry.

Examples

```
t <- c(3.4, NA, NA, NA, NA, 2.8, 2.1, 3.1, 2.0, 3.4)
n <- c(40, 20, 22, 24, 18, 30, 25, 30, 16, 22)
m <- meta(smc_from_t(t, n))
mean.age <- rnorm(10, 30, 5)
metalm(m, ~ mean.age)
```

smc_from_t

Calculate Effect Sizes for “meta.nsue” Objects

Description

These function calculate various effect sizes and output objects of class “nsue”, ready to be used by [meta](#) and [leave1out](#).

Usage

```
smc_from_t(t, n, alpha = 0.05, labels = "study", rm.r = 0.3)
smd_from_t(t, n1, n2, alpha = 0.05, labels = "study", rm.r = 0.3)
z_from_r(r, n, alpha = 0.05, labels = "study", rm.r = 0.3)
r_in_smd_from_t_means_and_sds1(t,
  n1, mean1.pre, sd1.pre, mean1.post, sd1.post,
  n2, mean2.pre, sd2.pre, mean2.post, sd2.post,
  alpha = 0.05, labels = "study", r.range = c(0, 0.99), rm.r = 0.3)
r_in_smd_from_t_means_and_sds2(x, maxiter = 200, tol = 1e-6)
```

Arguments

t	a vector to specify the t-values of the studies. Use NA in studies with Non Statistically-Significant Unreported Effects (NSUEs).
r	a vector to specify the correlation coefficients of the studies. Use NA in studies with Non Statistically-Significant Unreported Effects (NSUEs).
n	a vector to specify the sample sizes of the studies.
n1	a vector to specify the sample sizes of the first group (e.g. patients) of studies.

n2	a vector to specify the sample sizes of the second group (e.g. controls) of the studies.
mean1.pre	a vector to specify the means of the first group (e.g. patients) of the studies before the treatment.
sd1.pre	a vector to specify the standard deviations of the first group (e.g. patients) of the studies before the treatment.
mean1.post	a vector to specify the means of the first group (e.g. patients) of the studies after the treatment.
sd1.post	a vector to specify the standard deviations of the first group (e.g. patients) of the studies after the treatment.
mean2.pre	a vector to specify the means of the second group (e.g. patients) of the studies before the treatment.
sd2.pre	a vector to specify the standard deviations of the second group (e.g. patients) of the studies before the treatment.
mean2.post	a vector to specify the means of the second group (e.g. patients) of the studies after the treatment.
sd2.post	a vector to specify the standard deviations of the second group (e.g. patients) of the studies after the treatment.
alpha	a vector to specify the p-value thresholds used in the studies (e.g. 0.05).
labels	a vector to specify the names of the studies.
r.range	range of pre-post correlations.
rm.r	the expected correlation coefficient between repeated-measures.
x	an object of class "nsue".
maxiter	maximum number of iterations in the REML estimation of τ^2 .
tol	tolerance in the REML estimation of τ^2 .

Details

Use `smc_from_t` for calculating the standardized mean changes from the t-values of the paired Student t-tests, e.g. in repeated-measures studies analyzing the amount of change in within a group.

Use `smd_from_t` for calculating the standardized mean differences from t-values of the two-sample Student t-tests, e.g. in studies comparing a quantitative (normally-distributed) variable between two groups.

Use `z_from_r` for calculating the Fisher's r-to-z transformed correlations coefficients from the Pearson correlation coefficients (r), e.g. in studies examining the association between two quantitative (normally-distributed) variables.

Use `r_in_smd_from_t_means_and_sds1` and `meta` for estimating the missing pre-post correlations in a meta-analysis of the pre-post differences, e.g. when you only have the means and standard deviations before and after a treatment. Afterwards, use `r_in_smd_from_t_means_and_sds2` to conduct the meta-analysis of the pre-post differences. Please see Harrison et al for details.

Value

smc_from_t, smd_from_t, z_from_r and r_in_smd_from_t_means_and_sds1 return objects of class "nsue".

The function print may be used to print a summary of the results.

An object of class "nsue" is a list containing several of the following components:

measure	the effect-size measure used.
y	the effect-sizes.
y_lo	the effect-sizes corresponding to the lower statistical threshold.
y_up	the effect-sizes corresponding to the upper statistical threshold.
y.var	the variances of the effect sizes.
y2var_k1	a constant needed to derive the variances.
y2var_k2	a constant needed to derive the variances.
labels	the labels of the studies.
rm	a list with the expected correlation between repeated-measures studies.

Note

Studies with the same label will be considered to be repeated-measures, and will thus have lower weight during the MLE step, be imputed accounting for the correlation, and combined before meta-analysis.

Author(s)

Joaquim Radua

References

Radua, J., Schmidt, A., Borgwardt, S., Heinz, A., Schlagenhauf, F., McGuire, P., Fusar-Poli, P. (2015) Ventral striatal activation during reward processing in psychosis. A neurofunctional meta-analysis. *JAMA Psychiatry*, **72**, 1243–51.

Harrison, A., Fernandez de la Cruz, L., Enander, J., Radua, J., Mataix-Cols, D. (2016) Cognitive-behavioral therapy for body dysmorphic disorder: A systematic review and meta-analysis of randomized controlled trials. *Clinical Psychology Review*, in Press.

See Also

[meta](#) for conducting a meta-analysis.

[leave1out](#) for computing leave-one-out diagnostics.

Examples

```
t <- c(3.4, NA, NA, NA, NA, 2.8, 2.1, 3.1, 2.0, 3.4)
n <- c(40, 20, 22, 24, 18, 30, 25, 30, 16, 22)
x <- smc_from_t(t, n)
x
meta(x)
```

Index

- *Topic **anova**
 - linearHypothesis, 8
 - linearHypothesis.meta.nsue, 9
 - metalm, 15
 - metalm.meta.nsue, 16
- *Topic **bias**
 - funnel, 4
 - funnel.meta.nsue, 5
 - metabias, 13
 - metabias.meta.nsue, 14
- *Topic **comparison**
 - metalm, 15
 - metalm.meta.nsue, 16
- *Topic **effect-size**
 - smc_from_t, 17
- *Topic **hypothesis**
 - linearHypothesis, 8
 - linearHypothesis.meta.nsue, 9
- *Topic **jackknife**
 - leave1out, 6
 - leave1out.nsue, 7
- *Topic **leave1out**
 - leave1out, 6
 - leave1out.nsue, 7
- *Topic **meta-analysis**
 - forest, 2
 - forest.meta.nsue, 3
 - funnel, 4
 - funnel.meta.nsue, 5
 - leave1out, 6
 - leave1out.nsue, 7
 - meta, 10
 - meta.nsue, 11
 - metabias, 13
 - metabias.meta.nsue, 14
 - metalm, 15
 - metalm.meta.nsue, 16
- *Topic **plot**
 - forest, 2
 - forest.meta.nsue, 3
 - funnel, 4
 - funnel.meta.nsue, 5
- *Topic **regression**
 - linearHypothesis, 8
 - linearHypothesis.meta.nsue, 9
 - metabias, 13
 - metabias.meta.nsue, 14
 - metalm, 15
 - metalm.meta.nsue, 16
- *Topic **test**
 - linearHypothesis, 8
 - linearHypothesis.meta.nsue, 9
 - metabias, 13
 - metabias.meta.nsue, 14
 - metalm, 15
 - metalm.meta.nsue, 16
- coef.meta.nsue (meta.nsue), 11
- fitted.meta.nsue (meta.nsue), 11
- forest, 2, 4, 5, 10, 12
- forest.meta, 2
- forest.meta.nsue, 2, 3
- forest.rma, 2
- funnel, 2, 3, 4, 10, 12, 13, 15
- funnel.meta, 4
- funnel.meta.nsue, 4, 5
- funnel.rma, 4
- leave1out, 6, 10, 12, 17, 19
- leave1out.nsue, 6, 7, 11, 16
- leave1out.rma.uni, 6
- linearHypothesis, 2, 3, 8, 10, 12, 15–17
- linearHypothesis.default, 8
- linearHypothesis.meta.nsue, 8, 9, 9
- meta, 2–6, 8–10, 10, 13, 15–17, 19
- meta.nsue, 7, 10, 11, 16
- metabias, 4, 5, 10, 12, 13, 16, 17

metabias.meta, 13
metabias.meta.nsue, 13, 14
metalm, 8–10, 12, 13, 15, 15
metalm.meta.nsue, 7, 11, 15, 16, 16

pdf, 2–5
plot.meta.nsue (forest.meta.nsue), 3
print.leave1out.nsue (leave1out.nsue), 7
print.meta.nsue (meta.nsue), 11
print.nsue (smc_from_t), 17

r_in_smd_from_t_means_and_sds1, 6–8,
10–12
r_in_smd_from_t_means_and_sds1
(smc_from_t), 17
r_in_smd_from_t_means_and_sds2
(smc_from_t), 17
residuals.meta.nsue (meta.nsue), 11

smc_from_t, 6–8, 10–12, 17
smd_from_t, 6–8, 10–12
smd_from_t (smc_from_t), 17
subset.meta.nsue (meta.nsue), 11
summary.leave1out.nsue
(leave1out.nsue), 7
summary.meta.nsue (meta.nsue), 11

z_from_r, 6–8, 10–12
z_from_r (smc_from_t), 17