Package ‘metafor’

June 9, 2021

Version 3.0-2

Date 2021-06-09

Title Meta-Analysis Package for R

Depends R (>= 3.5.0), methods, Matrix

Imports stats, utils, graphics, grDevices, nlme, mathjaxr, pbapply

Suggests lme4, numDeriv, minqa, nloptr, dfoptim, ucminf, optimParallel, CompQuadForm, mvtnorm, BiasedUrn, Epi, survival, GLMMadaptive, multcomp, gsl, sp, ape, boot, crayon, R.rsp, testthat, rmarkdown

Description A comprehensive collection of functions for conducting meta-analyses in R. The package includes functions to calculate various effect sizes or outcome measures, fit fixed-, random-, and mixed-effects models to such data, carry out moderator and meta-regression analyses, and create various types of meta-analytical plots (e.g., forest, funnel, radial, L'Abbe, Baujat, bubble, and GOSH plots). For meta-analyses of binomial and person-time data, the package also provides functions that implement specialized methods, including the Mantel-Haenszel method, Peto's method, and a variety of suitable generalized linear (mixed-effects) models (i.e., mixed-effects logistic and Poisson regression models). Finally, the package provides functionality for fitting meta-analytic multivariate/multilevel models that account for non-independent sampling errors and/or true effects (e.g., due to the inclusion of multiple treatment studies, multiple endpoints, or other forms of clustering). Network meta-analyses and meta-analyses accounting for known correlation structures (e.g., due to phylogenetic relatedness) can also be conducted. An introduction to the package can be found in Viechtbauer (2010) <doi:10.18637/jss.v036.i03>.

License GPL (>=2)

ByteCompile TRUE

LazyData TRUE

Encoding UTF-8

RdMacros mathjaxr

VignetteBuilder R.rsp

URL https://www.metafor-project.org https://github.com/wviechtb/metafor https://wviechtb.github.io/metafor/ https://www.wvbauer.com

BugReports https://github.com/wviechtb/metafor/issues

R topics documented:

`metafor-package` *metafor: A Meta-Analysis Package for R*

Description

The **metafor** package provides a comprehensive collection of functions for conducting meta-analyses in R. The package can be used to calculate various effect size or outcome measures and then allows the user to fit fixed- and random-effects models to these data. By including study-level variables ('moderators') as predictors in these models, mixed-effects meta-regression models can also be fitted. For meta-analyses of 2×2 tables, proportions, incidence rates, and incidence rate ratios, the package also provides functions that implement specialized methods, including the Mantel-Haenszel method, Peto's method, and a variety of suitable generalized linear mixed-effects models (i.e., mixed-effects logistic and Poisson regression models). For non-independent effects/outcomes (e.g., due to correlated sampling errors, correlated true effects or outcomes, or other forms of clustering), the package also provides a function for fitting multilevel/multivariate meta-analytic models.

Various methods are available to assess model fit, to identify outliers and/or influential studies, and for conducting sensitivity analyses (e.g., standardized residuals, Cook's distances, leave-one-out analyses). Advanced techniques for hypothesis testing and obtaining confidence intervals (e.g., for the average effect or outcome or for the model coefficients in a meta-regression model) have also been implemented (e.g., the Knapp and Hartung method, permutation tests, cluster robust inference methods / robust variance estimation).

The package also provides functions for creating forest, funnel, radial (Galbraith), normal quantile-quantile, L'Abbé, Baujat, bubble, and GOSH plots. The presence of funnel plot asymmetry (which may be indicative of publication bias) and its potential impact on the results can be examined via the rank correlation and Egger's regression test, the trim and fill method, and by applying a variety of selection models.

The escalc Function

`escalc` Before a meta-analysis can be conducted, the relevant results from each study must be quantified in such a way that the resulting values can be further aggregated and compared. The `escalc` function can be used to compute a wide variety of effect size or outcome measures (and the corresponding sampling variances) that are often used in meta-analyses (e.g., risk ratios, odds ratios, risk differences, mean differences, standardized mean differences, response ratios / ratios of means, raw or r-to-z transformed correlation coefficients). Measures for quantifying some outcome for individual groups (e.g., proportions and incidence rates and transformations thereof), measures of change (e.g., raw and standardized mean changes), and measures of variability (e.g., variability ratios and coefficient of variation ratios) are also available.

The rma.uni Function

rma.uni The various meta-analytic models that are typically used in practice are special cases of the general linear (mixed-effects) model. The rma.uni function (with alias rma) provides a general framework for fitting such models. The function can be used in conjunction with any of the usual effect size or outcome measures used in meta-analyses (e.g., as computed using the escalc function). The notation and models underlying the rma.uni function are explained below.

For a set of $i = 1, \dots, k$ independent studies, let y_i denote the observed value of the effect size or outcome measure in the ith study. Let θ_i denote the corresponding (unknown) true effect/outcome, such that

$$y_i | \theta_i \sim N(\theta_i, v_i).$$

In other words, the observed effect sizes or outcomes are assumed to be unbiased and normally distributed estimates of the corresponding true effects/outcomes with sampling variances equal to v_i. The v_i values are assumed to be known. Depending on the outcome measure used, a bias correction, normalizing, and/or variance stabilizing transformation may be necessary to ensure that these assumptions are (approximately) true (e.g., the log transformation for odds/risk ratios, the bias correction for standardized mean differences, Fisher's r-to-z transformation for correlations; see escalc for more details).

The **fixed-effects model** conditions on the true effects/outcomes and therefore provides a *conditional inference* about the k studies included in the meta-analysis. When using weighted estimation, this implies that the fitted model provides an estimate of

$$\bar{\theta}_w = \sum_{i=1}^{k} w_i \theta_i / \sum_{i=1}^{k} w_i,$$

that is, the *weighted average* of the true effects/outcomes in the k studies, with weights equal to $w_i = 1/v_i$ (this is what is often described as the 'inverse-variance' method in the meta-analytic literature). One can also employ an unweighted estimation method, which provides an estimate of the *unweighted average* of the true effects/outcomes in k studies, that is, an estimate of

$$\bar{\theta}_u = \sum_{i=1}^{k} \theta_i / k.$$

For weighted estimation, one could also choose to estimate $\bar{\theta}_w$, where the w_i values are user-defined weights (inverse-variance weights or unit weights as in unweighted estimation are just special cases). It is up to the user to decide to what extent $\bar{\theta}_w$ is a meaningful parameter to estimate (regardless of the weights used).

The **random-effects model** does not condition on the true effects/outcomes. Instead, the k studies included in the meta-analysis are assumed to be a random sample from a larger population of studies. In rare cases, the studies included in a meta-analysis are actually sampled from a larger collection of studies. More typically, the population of studies is a hypothetical population of an essentially infinite set of studies comprising all of the studies that have been conducted, that could have been conducted, or that may be conducted in the future. We assume that $\theta_i \sim N(\mu, \tau^2)$, that is, the true effects/outcomes in the population of studies are normally distributed with μ denoting the average true effect/outcome and τ^2 the variance of the true effects/outcomes in the population (τ^2 is therefore often referred to as the amount of 'heterogeneity' in the true effects/outcomes). The random-effects model can also be written as

$$y_i = \mu + u_i + \epsilon_i,$$

where $u_i \sim N(0, \tau^2)$ and $\epsilon_i \sim N(0, v_i)$. The fitted model provides an estimate of μ and τ^2. Consequently, the random-effects model provides an *unconditional inference* about the average true effect/outcome in the population of studies (from which the k studies included in the meta-analysis are assumed to be a random sample).

When using weighted estimation in the context of a random-effects model, the model is fitted with weights equal to $w_i = 1/(\tau^2 + v_i)$, with τ^2 replaced by its estimate (this is the standard 'inverse-variance' method for random-effects models). One can also choose unweighted estimation in the context of the random-effects model or specify user-defined weights, although the parameter that is estimated (i.e., μ) remains the same regardless of the estimation method and weights used (as opposed to the fixed-effect model, where the parameter estimated is different for weighted versus unweighted estimation or when using different weights than the standard inverse-variance weights). Since weighted estimation with inverse-variance weights is most efficient, it is usually to be preferred for random-effects models (while in the fixed-effect model case, we must carefully consider whether $\bar{\theta}_w$ or $\bar{\theta}_u$ is the more meaningful parameter to estimate).

Contrary to what is often stated in the literature, it is important to realize that the fixed-effects model does *not* assume that the true effects/outcomes are homogeneous (i.e., that θ_i is equal to some common value θ in all k studies). In other words, fixed-effects models provide perfectly valid inferences under heterogeneity, as long as one is restricting these inferences to the set of studies included in the meta-analysis and one realizes that the model does not provide an estimate of θ, but of $\bar{\theta}_w$ or $\bar{\theta}_u$ (depending on the estimation method).

In the special case that the true effects/outcomes are actually homogeneous (the equal-effects case), the distinction between fixed- and random-effects models disappears, since homogeneity implies that $\mu = \bar{\theta}_w = \bar{\theta}_u \equiv \theta$. However, since there is no infallible method to test whether the true effects/outcomes are really homogeneous or not, a researcher should decide on the type of inference desired before examining the data and choose the model accordingly. In fact, there is nothing wrong with fitting both fixed- and random-effects models to the same data, since these models address different questions (i.e., what was the average effect/outcome in the studies that have been conducted versus what is the average effect/outcome in the larger population of studies?). For further details on the distinction between equal-, fixed-, and random-effects models, see Laird and Mosteller (1990) and Hedges and Vevea (1998).

Study-level variables (often referred to as 'moderators') can also be included as predictors in such models, leading to so-called 'meta-regression' analyses (to examine whether the effects/outcomes tend to be larger/smaller under certain conditions or circumstances). When including moderator variables in a random-effects model, we obtain a **mixed-effects meta-regression model**. This model can be written as

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_{p'} x_{ip'} + u_i + \epsilon_i,$$

where $u_i \sim N(0, \tau^2)$ and $\epsilon_i \sim N(0, v_i)$ as before and x_{ij} denotes the value of the jth moderator variable for the ith study (letting $p = p' + 1$ denote the total number of coefficients in the model including the model intercept). Therefore, β_j denotes how the average true effect/outcome changes for a one-unit increase in x_{ij} and the model intercept β_0 denotes the average true effect/outcome when the values of all moderator variables are equal to zero. The value of τ^2 in the mixed-effects model denotes the amount of 'residual heterogeneity' in the true effects/outcomes (i.e., the amount of variability in the true effects/outcomes that is not accounted for by the moderators included in the model).

The rma.mh Function

`rma.mh` The Mantel-Haenszel method provides an alternative approach for fitting fixed-effects models when dealing with studies providing data in the form of 2×2 tables or in the form of event counts (i.e., person-time data) for two groups (Mantel & Haenszel, 1959). The method is particularly advantageous when aggregating a large number of studies with small sample sizes (the so-called sparse data or increasing strata case). The Mantel-Haenszel method is implemented in the `rma.mh` function. It can be used in combination with risk ratios, odds ratios, risk differences, incidence rate ratios, and incidence rate differences.

The rma.peto Function

`rma.peto` Yet another method that can be used in the context of a meta-analysis of 2×2 table data is Peto's method (see Yusuf et al., 1985), implemented in the `rma.peto` function. The method provides an estimate of the (log) odds ratio under a fixed-effects model. The method is particularly advantageous when the event of interest is rare, but see the documentation of the function for some caveats.

The rma.glmm Function

`rma.glmm` Dichotomous outcomes and event counts (based on which one can calculate outcome measures such as odds ratios, incidence rate ratios, proportions, and incidence rates) are often assumed to arise from binomial and Poisson distributed data. Meta-analytic models that are directly based on such distributions are implemented in the `rma.glmm` function. These models are essentially special cases of generalized linear mixed-effects models (i.e., mixed-effects logistic and Poisson regression models). For 2×2 table data, a mixed-effects conditional logistic model (based on the non-central hypergeometric distribution) is also available. Random/mixed-effects models with dichotomous data are often referred to as 'binomial-normal' models in the meta-analytic literature. Analogously, for event count data, such models could be referred to as 'Poisson-normal' models.

The rma.mv Function

`rma.mv` Standard meta-analytic models assume independence between the observed effect sizes or outcomes obtained from a set of studies. This assumption is often violated in practice. Dependencies can arise for a variety of reasons. For example, the sampling errors and/or true effects/outcomes may be correlated in multiple treatment studies (e.g., when multiple treatment groups are compared with a common control/reference group, such that the data from the control/reference group is used multiple times to compute the observed effect sizes or outcomes) or in multiple endpoint studies (e.g., when more than one effect size estimate or outcome is calculated based on the same sample of subjects due to the use of multiple endpoints or response variables) (Gleser & Olkin, 2009). Correlations in the true effects/outcomes can also arise due to other forms of clustering (e.g., effects/outcomes derived from the same paper, lab, research group, or species may be more similar to each other than effects/outcomes derived from different papers, labs, research groups, or species). In ecology and related fields, shared phylogenetic history among the organisms studied (e.g., plants, fungi, animals) can also induce correlations among the effects/outcomes. The `rma.mv` function can be used to fit suitable meta-analytic multivariate/multilevel models to such data, so that the non-independence in the observed/true effects or outcomes is accounted for. Network meta-analyses (also called multiple/mixed treatment comparisons) can also be carried out with this function.

Future Plans and Updates

The **metafor** package is a work in progress and is updated on a regular basis with new functions and options. With `metafor.news()`, you can read the 'NEWS' file of the package after installation. Comments, feedback, and suggestions for improvements are always welcome.

Citing the Package

To cite the package, please use the following reference:

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Getting Started with the Package

The paper mentioned above is a good starting place for those interested in using the package. The purpose of the article is to provide a general overview of the package and its capabilities (as of version 1.4-0). Not all of the functions and options are described in the paper, but it should provide a useful introduction to the package. The paper can be freely downloaded from the URL given above or can be directly loaded with the command `vignette("metafor")`.

In addition to reading the paper, carefully read this page and then the help pages for the `escalc` and the `rma.uni` functions (or the `rma.mh`, `rma.peto`, `rma.glmm`, `rma.mv` functions if you intend to use these methods). The help pages for these functions provide links to many additional functions, which can be used after fitting a model. You can also read the entire documentation online at `https://wviechtb.github.io/metafor/reference/index.html` (where it is nicely formatted, equations are shown correctly, and the output from all examples is provided).

A (pdf) diagram showing the various functions in the metafor package (and how they are related to each other) can be opened with the command `vignette("diagram")`.

Finally, additional information about the package, several detailed analysis examples, examples of plots and figures provided by the package (with the corresponding code), some additional tips and notes, and a FAQ can be found on the package website at `https://www.metafor-project.org`.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>`
package website: `https://www.metafor-project.org`
author homepage: `https://www.wvbauer.com`

Suggestions on how to obtain help with using the package can found on the package website at: `https://www.metafor-project.org/doku.php/help`

References

Cooper, H., Hedges, L. V., & Valentine, J. C. (Eds.) (2009). *The handbook of research synthesis and meta-analysis* (2nd ed.). New York: Russell Sage Foundation.

Gleser, L. J., & Olkin, I. (2009). Stochastically dependent effect sizes. In H. Cooper, L. V. Hedges, & J. C. Valentine (Eds.), *The handbook of research synthesis and meta-analysis* (2nd ed., pp. 357–376). New York: Russell Sage Foundation.

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. San Diego, CA: Academic Press.

Hedges, L. V., & Vevea, J. L. (1998). Fixed- and random-effects models in meta-analysis. *Psychological Methods*, **3**(4), 486–504. https://doi.org/10.1037/1082-989X.3.4.486

Laird, N. M., & Mosteller, F. (1990). Some statistical methods for combining experimental results. *International Journal of Technology Assessment in Health Care*, **6**(1), 5–30. https://doi.org/10.1017/S0266462300008916

Mantel, N., & Haenszel, W. (1959). Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the National Cancer Institute*, **22**(4), 719–748. https://doi.org/10.1093/jnci/22.4.719

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Yusuf, S., Peto, R., Lewis, J., Collins, R., & Sleight, P. (1985). Beta blockade during and after myocardial infarction: An overview of the randomized trials. *Progress in Cardiovascular Disease*, **27**(5), 335–371. https://doi.org/10.1016/s0033-0620(85)80003-7

`addpoly` *Add Polygons to Forest Plots*

Description

The `addpoly` function can be used to add polygons, sometimes called ‘diamonds’, to a forest plot, for example to indicate summary estimates for subgroups of studies or to indicate fitted/predicted values based on models involving moderators.

Usage

```
addpoly(x, ...)
```

Arguments

`x`	either an object of class `"rma"` or the values at which polygons should be drawn. See ‘Details’.
`...`	other arguments.

Details

Currently, methods exist for two types of situations.

In the first case, object `x` is a fitted model coming from the `rma.uni`, `rma.mh`, `rma.peto`, or `rma.glmm` functions. The model must either be a fixed- or random-effects model, that is, the model should not contain any moderators. The corresponding method is called `addpoly.rma`. It can be used to add a polygon to an existing forest plot (usually at the bottom), showing the summary estimate (with its confidence interval) based on the fitted model.

Alternatively, object `x` can be a vector with values at which one or more polygons should be drawn. The corresponding method is then `addpoly.default`.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

addpoly.rma, addpoly.default, forest.rma, forest.default

---

addpoly.default *Add Polygons to Forest Plots (Default Method)*

Description

Function to add one or more polygons to a forest plot.

Usage

```
## Default S3 method:
addpoly(x, vi, sei, ci.lb, ci.ub, pi.lb, pi.ub,
        rows=-1, level=95, annotate=TRUE, digits=2, width, mlab,
        transf, atransf, targs, efac=1, col, border, fonts, cex, ...)
```

Arguments

`x` vector with the values at which the polygons should be drawn.

`vi` vector with the corresponding variances.

`sei` vector with the corresponding standard errors (note: only one of the two, `vi` or `sei`, needs to be specified).

`ci.lb` vector with the corresponding lower confidence interval bounds. Not needed if `vi` or `sei` is specified. See ‘Details’.

`ci.ub` vector with the corresponding upper confidence interval bounds. Not needed if `vi` or `sei` is specified. See ‘Details’.

`pi.lb` optional vector with the corresponding lower prediction interval bounds.

`pi.ub` optional vector with the corresponding upper prediction interval bounds.

`rows` vector to specify the rows (or more generally, the horizontal positions) for plotting the polygons (defaults is `-1`). Can also be a single value to specify the row (horizontal position) of the first polygon (the remaining polygons are then plotted below this starting row).

`level` numeric value between 0 and 100 to specify the confidence interval level (the default is 95).

`annotate` logical to specify whether annotations should be added to the plot for the polygons that are drawn (the default is `TRUE`).

`digits` integer to specify the number of decimal places to which the annotations should be rounded (the default is 2).

`width` optional integer to manually adjust the width of the columns for the annotations.

`mlab` optional character vector with the same length as `x` giving labels for the polygons that are drawn.

`transf` optional argument to specify a function that should be used to transform the `x` values and confidence interval bounds (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`atransf` optional argument to specify a function that should be used to transform the annotations (e.g., `atransf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified via `transf` or `atransf`.

`efac` vertical expansion factor for the polygons. The default value of 1 should usually work okay.

`col` optional character string to specify the name of a color to use for the polygons. If unspecified, the function sets a default color.

`border` optional character string to specify the name of a color to use for the border of the polygons. If unspecified, the function sets a default color.

`fonts` optional character string to specify the font to use for the labels and annotations. If unspecified, the default font is used.

`cex` optional symbol expansion factor. If unspecified, the function tries to set this to a sensible value.

`...` other arguments.

Details

The function can be used to add one or more polygons to an existing forest plot created with the `forest` function. For example, summary estimates based on a model involving moderators can be added to the plot this way (see 'Examples').

To use the function, one should specify the values at which the polygons should be drawn (via the `x` argument) together with the corresponding variances (via the `vi` argument) or with the corresponding standard errors (via the `sei` argument). Alternatively, one can specify the values at which the polygons should be drawn together with the corresponding confidence interval bounds (via the `ci.lb` and `ci.ub` arguments). Optionally, one can also specify the bounds of the corresponding prediction interval bounds via the `pi.lb` and `pi.ub` arguments.

The arguments `transf`, `atransf`, `efac`, and `cex` should always be set equal to the same values used to create the forest plot.

Author(s)

Wolfgang Viechtbauer <`wvb@metafor-project.org`> `https://www.metafor-project.org`

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

forest.rma, forest.default

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit mixed-effects model with absolute latitude as a moderator
res <- rma(yi, vi, mods = ~ ablat, slab=paste(author, year, sep=", "), data=dat)

### forest plot of the observed risk ratios
forest(res, addfit=FALSE, atransf=exp, xlim=c(-8,5), ylim=c(-4.5,16), cex=.8,
       order=dat$ablat, ilab=dat$ablat, ilab.xpos=-4,
       header="Author(s) and Year")

### predicted average log risk ratios for 10, 30, and 50 degrees absolute latitude
x <- predict(res, newmods=c(10, 30, 50))

### add predicted average risk ratios to forest plot
addpoly(x$pred, sei=x$se, atransf=exp, rows=-2,
        mlab=c("- at 10 Degrees", "- at 30 Degrees", "- at 50 Degrees"), cex=.8)
abline(h=0)
text(-8, -1, "Model-Based Estimates:", pos=4, cex=.8)
text(-4, 15, "Latitude", cex=.8, font=2)
```

---

addpoly.rma *Add Polygons to Forest Plots (Method for 'rma' Objects)*

Description

Function to add a polygon to a forest plot showing the summary estimate with corresponding confidence interval based on an object of class "rma".

Usage

```
## S3 method for class 'rma'
addpoly(x, row=-2, level=x$level, annotate=TRUE,
        addpred=FALSE, digits=2, width, mlab, transf, atransf, targs,
        efac=1, col, border, fonts, cex, ...)
```

Arguments

`x` an object of class `"rma"`.

`row` numeric value to specify the row (or more generally, the horizontal position) for plotting the polygon (the default is `-2`).

`level` numeric value between 0 and 100 to specify the confidence interval level (the default is to take the value from the object).

`annotate` logical to specify whether annotations for the summary estimate should be added to the plot (the default is `TRUE`).

`addpred` logical to specify whether the bounds of the prediction interval should be added to the plot (the default is `FALSE`).

`digits` integer to specify the number of decimal places to which the annotations should be rounded (the default is 2).

`width` optional integer to manually adjust the width of the columns for the annotations.

`mlab` optional character string giving a label for the summary estimate polygon. If unspecified, the function sets a default label.

`transf` optional argument to specify a function that should be used to transform the summary estimate and confidence interval bound (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`atransf` optional argument to specify a function that should be used to transform the annotations (e.g., `atransf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified via `transf` or `atransf`.

`efac` vertical expansion factor for the polygon. The default value of 1 should usually work okay.

`col` optional character string to specify the name of a color to use for the polygon. If unspecified, the function sets a default color.

`border` optional character string to specify the name of a color to use for the border of the polygon. If unspecified, the function sets a default color.

`fonts` optional character string to specify the font to use for the label and annotations. If unspecified, the default font is used.

`cex` optional symbol expansion factor. If unspecified, the function tries to set this to a sensible value.

`...` other arguments.

Details

The function can be used to add a four-sided polygon, sometimes called a summary ‘diamond’, to an existing forest plot created with the `forest` function. The polygon shows the summary estimate (with its confidence interval bounds) based on a fixed- or random-effects model. Using this function, summary estimates based on different types of models can be shown in the same plot. Also, summary estimates based on a subgrouping of the studies can be added to the plot this way. See ‘Examples’.

The arguments `transf`, `atransf`, `efac`, and `cex` should always be set equal to the same values used to create the forest plot.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

forest.rma, forest.default

Examples

```
### meta-analysis of the log risk ratios using the Mantel-Haenszel method
res <- rma.mh(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg,
              slab=paste(author, year, sep=", "))

### forest plot of the observed risk ratios with summary estimate
forest(res, atransf=exp, xlim=c(-8,6), ylim=c(-2.5,16), header=TRUE)

### meta-analysis of the log risk ratios using a random-effects model
res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### add summary estimate from the random-effects model to the forest plot
addpoly(res, atransf=exp)

### forest plot with subgrouping of studies and summaries per subgroup
res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg,
           slab=paste(author, year, sep=", "))
forest(res, xlim=c(-16, 4.6), at=log(c(.05, .25, 1, 4)), atransf=exp,
       ilab=cbind(dat.bcg$tpos, dat.bcg$tneg, dat.bcg$cpos, dat.bcg$cneg),
       ilab.xpos=c(-9.5,-8,-6,-4.5), cex=.75, ylim=c(-1, 27),
       order=dat.bcg$alloc, rows=c(3:4,9:15,20:23),
       mlab="RE Model for All Studies", header="Author(s) and Year")
op <- par(cex=.75, font=2)
text(c(-9.5,-8,-6,-4.5), 26, c("TB+", "TB-", "TB+", "TB-"))
text(c(-8.75,-5.25),     27, c("Vaccinated", "Control"))
par(font=4)
text(-16, c(24,16,5), c("Systematic Allocation", "Random Allocation",
                        "Alternate Allocation"), pos=4)
par(op)
res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg,
           subset=(alloc=="systematic"))
addpoly(res, row=18.5, cex=.75, atransf=exp, mlab="RE Model for Subgroup")
res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg,
           subset=(alloc=="random"))
addpoly(res, row=7.5, cex=.75, atransf=exp, mlab="RE Model for Subgroup")
res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg,
           subset=(alloc=="alternate"))
addpoly(res, row=1.5, cex=.75, atransf=exp, mlab="RE Model for Subgroup")
```

`aggregate.escalc` *Aggregate Multiple Effect Sizes or Outcomes Within Studies*

Description

The function can be used to aggregate multiple effect sizes or outcomes belonging to the same study (or to the same level of some other clustering variable) into a single combined effect size or outcome.

Usage

```
## S3 method for class 'escalc'
aggregate(x, cluster, time, V, struct="CS", rho, phi,
          weighted=TRUE, fun, na.rm=TRUE, subset, select, digits, ...)
```

Arguments

`x` an object of class `"escalc"`.

`cluster` vector to specify the clustering variable (e.g., study).

`time` optional vector to specify the time points (only relevant when `struct="CAR"` or `struct="CS+CAR"`).

`V` optional argument to specify the variance-covariance matrix of the sampling errors. If not specified, argument `struct` is used to specify the variance-covariance structure.

`struct` character string to specify the variance-covariance structure of the sampling errors within the same cluster (either `"ID"`, `"CS"`, `"CAR"`, or `"CS+CAR"`). See 'Details'.

`rho` value of the correlation of the sampling errors within clusters (when `struct="CS"` or `struct="CS+CAR"`). Can also be a vector with the value of the correlation for each cluster.

`phi` value of the autocorrelation of the sampling errors within clusters (when `struct="CAR"` or `struct="CS+CAR"`). Can also be a vector with the value of the autocorrelation for each cluster.

`weighted` logical to specify whether estimates within clusters should be aggregated using inverse-variance weighting (the default is `TRUE`). If set to `FALSE`, unweighted averages are computed.

`fun` optional list with three functions for aggregating other variables besides the effect sizes or outcomes within clusters (for numeric/integer variables, for logicals, and for all other types, respectively).

`na.rm` logical to specify whether `NA` values should be removed before aggregating values within clusters. Can also be a vector with two logicals (the first pertains to the effect sizes or outcomes, the second to all other variables).

`subset` optional (logical or numeric) vector to specify the subset of rows to include when aggregating the effect sizes or outcomes.

select optional vector to specify the names of the variables to include in the aggregated dataset.

digits integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).

... other arguments.

Details

In many meta-analyses, multiple effect sizes or outcomes can be extracted from the same study. Ideally, such structures should be analyzed using an appropriate multilevel/multivariate model as can be fitted with the rma.mv function. However, there may occasionally be reasons for aggregating multiple effect sizes or outcomes belonging to the same study (or to the same level of some other clustering variable) into a single combined effect size or outcome. The present function can be used for this purpose.

The input must be an object of class "escalc". The error 'Error in match.fun(FUN): argument "FUN" is missing,with no default' indicates that a regular data frame was passed to the function, but this does not work. One can turn a regular data frame (containing the effect sizes or outcomes and the corresponding sampling variances) into an "escalc" object with the escalc function. See the 'Examples' below for an illustration of this.

The cluster variable is used to specify which estimates/outcomes belong to the same study/cluster.

In the simplest case, the estimates/outcomes within clusters (or, to be precise, their sampling errors) are assumed to be independent. This is usually a safe assumption as long as each study participant (or whatever the unit of analysis is) only contributes data to a single estimate/outcome. For example, if a study provides effect size estimates for male and female subjects separately, then the sampling errors can usually be assumed to be independent. In this case, one can set struct="ID" and multiple estimates/outcomes within the same cluster are combined using standard inverse-variance weighting (i.e., using weighted least squares) under the assumption of independence.

In other cases, the estimates/outcomes within clusters cannot be assumed to be independent. For example, if multiple effect size estimates are computed for the same group of subjects (e.g., for different dependent variables), then the estimates are likely to be correlated. If the actual correlation between the estimates is unknown, one can often still make an educated guess and set argument rho to this value, which is then assumed to be the same for all pairs of estimates within clusters when struct="CS" (for a compound symmetric structure). Multiple estimates/outcomes within the same cluster are then combined using inverse-variance weighting taking their correlation into consideration (i.e., using generalized least squares). One can also specify a different value of rho for each cluster by passing a vector (of the same length as the number of clusters) to this argument.

If multiple effect size estimates are computed for the same group of subjects at different time points, then it may be more sensible to assume that the correlation between estimates decreases as a function of the distance between the time points. If so, one can specify struct="CAR" (for a continuous-time autoregressive structure), set phi to the autocorrelation (for two estimates one time-unit apart), and use argument time to specify the actual time points corresponding to the estimates. The correlation between two estimates, y_{ij} and $y_{ij'}$, in the ith cluster, with time points t_{ij} and $t_{ij'}$, is then given by $\phi^{|t_{ij}-t_{ij'}|}$. One can also specify a different value of phi for each cluster by passing a vector (of the same length as the number of clusters) to this argument.

One can also combine the compound symmetric and autoregressive structures by specifying struct="CS+CAR". In this case, one must specify both rho and phi. The correlation between two estimates, y_{ij} and $y_{ij'}$, in the ith cluster, with time points t_{ij} and $t_{ij'}$, is then given by $\rho + (1-\rho)\phi^{|t_{ij}-t_{ij'}|}$.

Finally, if one actually knows the correlation (and hence the covariance) between each pair of estimates, one can also specify the entire variance-covariance matrix of the estimates (or more precisely, their sampling errors) via the V argument. In this case, arguments struct, rho, and phi are ignored.

Instead of using inverse-variance weighting (i.e., weighted/generalized least squares) to combine the estimates within clusters, one can set weighted=FALSE in which case the estimates are averaged within clusters without any weighting.

Other variables (besides the estimates) will also be aggregated to the cluster level. By default, numeric/integer type variables are averaged, logicals are also averaged (yielding the proportion of TRUE values), and for all other types of variables (e.g., character variables or factors) the most frequent category/level is returned. One can also specify a list of three functions via the fun argument for aggregating variables belong to these three types.

Argument na.rm controls how missing values should be handled. By default, any missing estimates are first removed before aggregating the non-missing values within each cluster. The same applies when aggregating the other variables. One can also specify a vector with two logicals for the na.rm argument to control how missing values should be handled when aggregating the estimates and when aggregating all other variables.

Value

An object of class c("escalc","data.frame") that contains the (selected) variables aggregated to the cluster level.

The object is formatted and printed with the print.escalc function.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

escalc

Examples

```
### copy data into 'dat' and examine data
dat <- dat.konstantopoulos2011
dat

### aggregate estimates to the district level, assuming independent sampling
### errors for multiples studies/schools within the same district
agg <- aggregate(dat, cluster=district, struct="ID")
agg

### copy data into 'dat' and examine data
```

```
dat <- dat.assink2016
dat

### note: 'dat' is a regular data frame
class(dat)

### turn data frame into an 'escalc' object
dat <- escalc(yi=yi, vi=vi, data=dat)
class(dat)

### aggregate the estimates to the study level, assuming a CS structure for
### the sampling errors within studies with a correlation of 0.6
agg <- aggregate(dat, cluster=study, rho=0.6)
agg

### reshape 'dat.ishak2007' into long format
dat <- dat.ishak2007
dat <- reshape(dat.ishak2007, direction="long", idvar="study", v.names=c("yi","vi"),
                    varying=list(c(2,4,6,8), c(3,5,7,9)))
dat <- dat[order(dat$study, dat$time),]
is.miss  <- is.na(dat$yi)
dat <- dat[!is.miss,]
rownames(dat) <- NULL
dat

### aggregate the estimates to the study level, assuming a CAR structure for
### the sampling errors within studies with an autocorrelation of 0.9
agg <- aggregate(dat, cluster=study, struct="CAR", time=time, phi=0.9)
agg
```

anova.rma *Likelihood Ratio and Wald-Type Tests for 'rma' Objects*

Description

For two (nested) models of class "rma.uni" or "rma.mv", the function provides a full versus reduced model comparison in terms of model fit statistics and a likelihood ratio test. When a single model is specified, a Wald-type test of one or more model coefficients or linear combinations thereof is carried out.

Usage

```
## S3 method for class 'rma'
anova(object, object2, btt, X, att, Z, digits, ...)
```

Arguments

object	an object of class "rma.uni" or "rma.mv".
object2	an (optional) object of class "rma.uni" or "rma.mv". See 'Details'.

`btt` optional vector of indices to specify which coefficients should be included in the Wald-type test. Can also be a string to grep for. See 'Details'.

`X` optional numeric vector or matrix to specify one or more linear combinations of the coefficients in the model that should be tested. See 'Details'.

`att` optional vector of indices to specify which scale coefficients should be included in the Wald-type test. Can also be a string to grep for. See 'Details'. Only relevant for location-scale models (see `rma`).

`Z` optional numeric vector or matrix to specify one or more linear combinations of the scale coefficients in the model that should be tested. See 'Details'. Only relevant for location-scale models (see `rma`).

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.

`...` other arguments.

Details

When a single model is specified, the function provides Wald-type tests of one or more model coefficients or linear combinations thereof.

In particular, for a fixed- or random-effects model (i.e., a model without moderators), this is just the test of the single coefficient of the model. For models including moderators, an omnibus test of all the model coefficients is conducted that excludes the intercept (the first coefficient) if it is included in the model. If no intercept is included in the model, then the omnibus test includes all of the coefficients in the model including the first.

Alternatively, one can manually specify the indices of the coefficients to test via the `btt` argument. For example, with `btt=c(3,4)`, only the third and fourth coefficient from the model would be included in the test (if an intercept is included in the model, then it corresponds to the first coefficient in the model). Instead of specifying the coefficient numbers, one can specify a string for `btt`. In that case, `grep` will be used to search for all coefficient names that match the string. Using the `btt` argument, one can for example select all coefficients corresponding to a particular factor to test if the factor as a whole is significant. See 'Examples'.

Instead, one can use the `X` argument to specify a linear combination of the coefficients in the model that should be tested (i.e., whether the linear combination is significantly different from zero). If a matrix of linear combinations is specified, each row defines a particular linear combination to be tested. If the matrix is of full rank, an omnibus Wald-type test of all linear combinations is also provided. Linear combinations can also be obtained with the `predict.rma` function, which will provide corresponding confidence intervals.

When specifying two models for comparison, the function provides a likelihood ratio test (LRT) comparing the two models. The two models must be based on the same set of data, must be of the same class, and should be nested for the LRT to make sense. Note that LRTs are not meaningful when using REML estimation and the two models differ in their fixed effects.

For location-scale models fitted with the `rma` function, one can use `att` to specify the indices of the scale coefficients to test. Similarly, one can use the `Z` argument to specify one or multiple linear combinations of the scale coefficients in the model that should be tested.

Value

An object of class "anova.rma". When a single model is specified (without any further arguments or together with the btt argument), the object is a list containing the following components:

QM test statistic of the Wald-type test of the model coefficients.

QMdf corresponding degrees of freedom.

QMp corresponding p-value.

btt indices of the coefficients tested by the Wald-type test.

k number of outcomes included in the analysis.

p number of coefficients in the model (including the intercept).

m number of coefficients included in the Wald-type test.

... some additional elements/values.

When argument X is used, the object is a list containing the following components:

QM test statistic of the omnibus Wald-type test of all linear combinations.

QMdf corresponding degrees of freedom.

QMp corresponding p-value.

hyp description of the linear combinations tested.

Xb values of the linear combinations.

se standard errors of the linear combinations.

zval test statistics of the linear combinations.

pval corresponding p-values.

When two models are specified, the object is a list containing the following components:

fit.stats.f log-likelihood, deviance, AIC, BIC, and AICc for the full model.

fit.stats.r log-likelihood, deviance, AIC, BIC, and AICc for the reduced model.

parms.f number of parameters in the full model.

parms.r number of parameters in the reduced model.

LRT likelihood ratio test statistic.

pval corresponding p-value.

QE.f test statistic of the test for (residual) heterogeneity from the full model.

QE.r test statistic of the test for (residual) heterogeneity from the reduced model.

tau2.f estimated τ^2 value from the full model. NA for "rma.mv" objects.

tau2.r estimated τ^2 value from the reduced model. NA for "rma.mv" objects.

R2 amount (in percent) of the heterogeneity in the reduced model that is accounted for in the full model (NA for fixed-effects models or for "rma.mv" objects). This can be regarded as a pseudo R^2 statistic (Raudenbush, 2009). Note that the value may not be very accurate unless k is large (Lopez-Lopez et al., 2014).

... some additional elements/values.

The results are formatted and printed with the print.anova.rma function.

Note

The function can also be used to conduct a likelihood ratio test (LRT) for the amount of (residual) heterogeneity in random- and mixed-effects models. The full model should then be fitted with either `method="ML"` or `method="REML"` and the reduced model with `method="FE"`. The p-value for the test is based on a chi-square distribution with 1 degree of freedom, but actually needs to be adjusted for the fact that the parameter (i.e., τ^2) falls on the boundary of the parameter space under the null hypothesis (see Viechtbauer, 2007, for more details).

LRTs for variance components in more complex models (as fitted with the `rma.mv` function) can also be conducted in this manner (see 'Examples').

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Hardy, R. J., & Thompson, S. G. (1996). A likelihood approach to meta-analysis with random effects. *Statistics in Medicine*, **15**(6), 619–629. https://doi.org/10.1002/(sici)1097-0258(19960330)15:6%3C619::aid-sim188%3E3.0.co;2-a

Huizenga, H. M., Visser, I., & Dolan, C. V. (2011). Testing overall and moderator effects in random effects meta-regression. *British Journal of Mathematical and Statistical Psychology*, **64**(1), 1–19. https://doi.org/10.1348/000711010X522687

López-López, J. A., Marín-Martínez, F., Sánchez-Meca, J., Van den Noortgate, W., & Viechtbauer, W. (2014). Estimation of the predictive power of the model in mixed-effects meta-regression: A simulation study. *British Journal of Mathematical and Statistical Psychology*, **67**(1), 30–48. https://doi.org/10.1111/bmsp.12002

Raudenbush, S. W. (2009). Analyzing effect sizes: Random effects models. In H. Cooper, L. V. Hedges, & J. C. Valentine (Eds.), *The handbook of research synthesis and meta-analysis* (2nd ed., pp. 295–315). New York: Russell Sage Foundation.

Viechtbauer, W. (2007). Hypothesis tests for population heterogeneity in meta-analysis. *British Journal of Mathematical and Statistical Psychology*, **60**(1), 29–60. https://doi.org/10.1348/000711005X64042

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`rma.uni`, `rma.mv`, `print.anova.rma`

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit random-effects model
res1 <- rma(yi, vi, data=dat, method="ML")

### fit mixed-effects model with two moderators (absolute latitude and publication year)
res2 <- rma(yi, vi, mods = ~ ablat + year, data=dat, method="ML")
```

```
### Wald-type test of the two moderators
anova(res2)

### alternative way of specifying the same test
anova(res2, X=rbind(c(0,1,0), c(0,0,1)))

### corresponding likelihood ratio test
anova(res1, res2)

### test of a linear combination
anova(res2, X=c(1,35,1970))

### use predict() to obtain the same linear combination (with its CI)
predict(res2, newmods=c(35,1970))

### mixed-effects model with three moderators
res3 <- rma(yi, vi, mods = ~ ablat + year + alloc, data=dat, method="ML")
res3

### test the 'alloc' factor
anova(res3, btt=4:5)

### instead of specifying the coefficient numbers, grep for "alloc"
anova(res3, btt="alloc")

############################################################################

### an example of doing LRTs of variance components in more complex models
dat <- dat.konstantopoulos2011
res <- rma.mv(yi, vi, random = ~ 1 | district/school, data=dat)

### test the district-level variance component
res0 <- rma.mv(yi, vi, random = ~ 1 | district/school, data=dat, sigma2=c(0,NA))
anova(res, res0)

### test the school-level variance component
res0 <- rma.mv(yi, vi, random = ~ 1 | district/school, data=dat, sigma2=c(NA,0))
anova(res, res0)

### test both variance components simultaneously
res0 <- rma.mv(yi, vi, data=dat)
anova(res, res0)
```

`baujat` *Baujat Plots for 'rma' Objects*

Description

Function to create Baujat plots for objects of class `"rma"`.

Usage

```
baujat(x, ...)

## S3 method for class 'rma'
baujat(x, xlim, ylim, xlab, ylab, cex, symbol="ids", grid=TRUE, progbar=FALSE, ...)
```

Arguments

`x` an object of class `"rma"`.

`xlim` x-axis limits. If unspecified, the function tries to set the x-axis limits to some sensible values.

`ylim` y-axis limits. If unspecified, the function tries to set the y-axis limits to some sensible values.

`xlab` title for the x-axis. If unspecified, the function tries to set an appropriate axis title.

`ylab` title for the y-axis. If unspecified, the function tries to set an appropriate axis title.

`cex` optional character expansion factor. If unspecified, the function tries to set this to a sensible value.

`symbol` either an integer to specify the `pch` value (i.e., plotting symbol), or `"slab"` to plot the study labels, or `"ids"` (the default) to plot the study id numbers.

`grid` logical to specify whether a grid should be added to the plot (can also be a color name).

`progbar` logical to specify whether a progress bar should be shown (the default is `FALSE`).

`...` other arguments.

Details

The model specified via `x` must be a model fitted with either the `rma.uni`, `rma.mh`, or `rma.peto` functions.

Baujat et al. (2002) proposed a diagnostic plot to detect sources of heterogeneity in meta-analytic data. The plot shows the contribution of each study to the overall Q-test statistic for heterogeneity on the x-axis versus the influence of each study (defined as the standardized squared difference between the overall estimate based on a fixed-effects model with and without the study included in the model fitting) on the y-axis. The same type of plot can be produced by first fitting a fixed-effects model with either the `rma.uni` (using `method="FE"`), `rma.mh`, or `rma.peto` functions and then passing the fitted model object to the `baujat` function.

For models fitted with the `rma.uni` function (which may involve moderators and/or which may be random/mixed-effects models), the idea underlying this type of plot can be generalized as follows (Viechtbauer, 2021): The x-axis then corresponds to the squared Pearson residual of a study, while the y-axis corresponds to the standardized squared difference between the predicted/fitted value for the study with and without the study included in the model fitting. Therefore, for a fixed-effects with moderators model, the x-axis corresponds to the contribution of the study to the Q_E-test statistic for residual heterogeneity.

By default, the points plotted are the study id numbers, but one can also plot the study labels by setting `symbol="slab"` (if study labels are available within the model object) or one can specify a plotting symbol via the `symbol` argument that gets passed to `pch` (see `points` for possible options).

Value

A data frame with components:

`x`	the x-axis coordinates of the points that were plotted.
`y`	the y-axis coordinates of the points that were plotted.
`ids`	the study id numbers.
`slab`	the study labels.

Note that the data frame is returned invisibly.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Baujat, B., Mahe, C., Pignon, J.-P., & Hill, C. (2002). A graphical method for exploring heterogeneity in meta-analyses: Application to a meta-analysis of 65 trials. *Statistics in Medicine*, **21**(18), 2641–2652. https://doi.org/10.1002/sim.1221

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Viechtbauer, W. (2021). Model checking in meta-analysis. In C. H. Schmid, T. Stijnen, & I. R. White (Eds.), *Handbook of meta-analysis* (pp. 219–254). Boca Raton, FL: CRC Press. https://doi.org/10.1201/978131511940

See Also

`rma.uni`, `rma.mh`, `rma.peto`, `influence.rma.uni`, `radial`

Examples

```
### copy data from Pignon et al. (2000) into 'dat'
dat <- dat.pignon2000

### calculate estimated log hazard ratios and sampling variances
dat$yi <- with(dat, OmE/V)
dat$vi <- with(dat, 1/V)

### meta-analysis based on all 65 trials
res <- rma(yi, vi, data=dat, method="FE", slab=trial)

### create Baujat plot
baujat(res)

### some variations of the plotting symbol
baujat(res, symbol=19)
baujat(res, symbol="slab")

### label only a selection of the more 'extreme' points
sav <- baujat(res, symbol=19, xlim=c(0,20))
sav <- sav[sav$x >= 10 | sav$y >= 0.10,]
text(sav$x, sav$y, sav$slab, pos=1)
```

---

bldiag *Construct Block Diagonal Matrix*

Description

Function to construct a block diagonal matrix from (a list of) matrices.

Usage

```
bldiag(..., order)
```

Arguments

`...` individual matrices or a list of matrices.

`order` optional argument to specify a variable based on which a square block diagonal matrix should be ordered.

Author(s)

Posted to R-help by Berton Gunter (2 Sep 2005) with some further adjustments by Wolfgang Viechtbauer

See Also

rma.mv

Examples

```
### copy data into 'dat'
dat <- dat.berkey1998
dat

### construct list with the variance-covariance matrices of the observed outcomes for the studies
V <- lapply(split(dat[c("v1i", "v2i")], dat$trial), as.matrix)
V

### construct block diagonal matrix
V <- bldiag(V)
V

### if we split based on 'author', the list elements in V are in a different order than tha data
V <- lapply(split(dat[c("v1i", "v2i")], dat$author), as.matrix)
```

```
V

### can use 'order' argument to reorder the block-diagonal matrix into the correct order
V <- bldiag(V, order=dat$author)
V
```

blup *Best Linear Unbiased Predictions for 'rma.uni' Objects*

Description

The function calculates best linear unbiased predictions (BLUPs) of the study-specific true effect sizes or outcomes by combining the fitted values based on the fixed effects and the estimated contributions of the random effects for objects of class "rma.uni". Corresponding standard errors and prediction interval bounds are also provided.

Usage

```
blup(x, ...)

## S3 method for class 'rma.uni'
blup(x, level, digits, transf, targs, ...)
```

Arguments

x	an object of class "rma.uni".
level	numeric value between 0 and 100 to specify the prediction interval level. If unspecified, the default is to take the value from the object.
digits	integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.
transf	optional argument to specify a function that should be used to transform the predicted values and interval bounds (e.g., transf=exp; see also transf). If unspecified, no transformation is used.
targs	optional arguments needed by the function specified under transf.
...	other arguments.

Value

An object of class "list.rma". The object is a list containing the following components:

pred	predicted values.
se	corresponding standard errors.
pi.lb	lower bound of the prediction intervals.
pi.ub	upper bound of the prediction intervals.
...	some additional elements/values.

The object is formatted and printed with print.list.rma.

Note

For best linear unbiased predictions of only the random effects, see `ranef`.

For predicted/fitted values that are based only on the fixed effects of the model, see `fitted.rma` and `predict.rma`.

For conditional residuals (the deviations of the observed effect sizes or outcomes from the BLUPs), see `rstandard.rma.uni` with `type="conditional"`.

Fixed-effects models (with or without moderators) do not contain random study effects. The BLUPs for these models will therefore be equal to the fitted values, that is, those obtained with `fitted.rma` and `predict.rma`.

When using the `transf` argument, the transformation is applied to the predicted values and the corresponding interval bounds. The standard errors are then set equal to `NA` and are omitted from the printed output.

By default, a standard normal distribution is used to calculate the prediction intervals. When the model was fitted with `test="t"` or `test="knha"`, then a t-distribution with $k-p$ degrees of freedom is used.

To be precise, it should be noted that the function actually calculates empirical BLUPs (eBLUPs), since the predicted values are a function of the estimated value of τ^2.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Kackar, R. N., & Harville, D. A. (1981). Unbiasedness of two-stage estimation and prediction procedures for mixed linear models. Communications in Statistics, Theory and Methods, **10**(13), 1249–1261. https://doi.org/10.1080/03610928108828108

Raudenbush, S. W., & Bryk, A. S. (1985). Empirical Bayes meta-analysis. *Journal of Educational Statistics*, **10**(2), 75–98. https://doi.org/10.3102/10769986010002075

Robinson, G. K. (1991). That BLUP is a good thing: The estimation of random effects. *Statistical Science*, **6**(1), 15–32. https://doi.org/10.1214/ss/1177011926

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`rma.uni`, `predict.rma`, `fitted.rma`, `ranef.rma.uni`

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### meta-analysis of the log risk ratios using a random-effects model
res <- rma(yi, vi, data=dat)
```

```
### BLUPs of the true risk ratios for each study
blup(res, transf=exp)

### illustrate shrinkage of BLUPs towards the (estimated) population average
res <- rma(yi, vi, data=dat)
blups <- blup(res)$pred
plot(NA, NA, xlim=c(.8,2.4), ylim=c(-2,0.5), pch=19,
     xaxt="n", bty="n", xlab="", ylab="Log Risk Ratio")
segments(rep(1,13), dat$yi, rep(2,13), blups, col="darkgray")
points(rep(1,13), dat$yi, pch=19)
points(rep(2,13), blups, pch=19)
axis(side=1, at=c(1,2), labels=c("Observed\nValues", "BLUPs"), lwd=0)
segments(.7, res$beta, 2.15, res$beta, lty="dotted")
text(2.3, res$beta, expression(hat(mu)==-0.71), cex=1)
```

`coef.permutest.rma.uni`

Extract the Model Coefficient Table from ’permutest.rma.uni’ Objects

---

Description

The function extracts the estimated model coefficients, corresponding standard errors, test statistics, p-values (based on the permutation tests), and confidence interval bounds from objects of class `"permutest.rma.uni"`.

Usage

```
## S3 method for class 'permutest.rma.uni'
coef(object, ...)
```

Arguments

`object`	an object of class `"permutest.rma.uni"`.
`...`	other arguments.

Value

A data frame with the following elements:

`estimate`	estimated model coefficient(s).
`se`	corresponding standard error(s).
`zval`	corresponding test statistic(s).
`pval`	p-value(s) based on the permutation test(s).
`ci.lb`	lower bound of the (permutation-based) confidence interval(s).
`ci.ub`	upper bound of the (permutation-based) confidence interval(s).

When the model was fitted with `test="t"` or `test="knha"`, then `zval` is called `tval` in the data frame that is returned by the function.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

permutest.rma.uni, rma.uni

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)

### carry out permutation test
## Not run:
sav <- permutest(res)
coef(sav)
## End(Not run)
```

---

coef.rma *Extract the Model Coefficients and Coefficient Table from 'rma' and 'summary.rma' Objects*

Description

The coef function extracts the estimated model coefficients from objects of class "rma". For objects of class "summary.rma", the model coefficients, corresponding standard errors, test statistics, p-values, and confidence interval bounds are extracted.

Usage

```
## S3 method for class 'rma'
coef(object, ...)
## S3 method for class 'summary.rma'
coef(object, ...)
```

Arguments

object	an object of class "rma" or "summary.rma".
...	other arguments.

Value

Either a vector with the estimated model coefficient(s) or a data frame with the following elements:

`estimate`	estimated model coefficient(s).
`se`	corresponding standard error(s).
`zval`	corresponding test statistic(s).
`pval`	corresponding p-value(s).
`ci.lb`	corresponding lower bound of the confidence interval(s).
`ci.ub`	corresponding upper bound of the confidence interval(s).

When the model was fitted with `test="t"` or `test="knha"`, then `zval` is called `tval` in the data frame that is returned by the function.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`rma.uni`, `rma.mh`, `rma.peto`, `rma.glmm`, `rma.mv`

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)

### extract model coefficients
coef(res)

### extract model coefficient table
coef(summary(res))
```

confint.rma *Confidence Intervals for 'rma' Objects*

Description

The function calculates confidence intervals for the model coefficients and/or other parameters in the model.

Usage

```
## S3 method for class 'rma.uni'
confint(object, parm, level, fixed=FALSE, random=TRUE, type,
        digits, transf, targs, verbose=FALSE, control, ...)

## S3 method for class 'rma.mh'
confint(object, parm, level, digits, transf, targs, ...)

## S3 method for class 'rma.peto'
confint(object, parm, level, digits, transf, targs, ...)

## S3 method for class 'rma.glmm'
confint(object, parm, level, digits, transf, targs, ...)

## S3 method for class 'rma.mv'
confint(object, parm, level, fixed=FALSE, sigma2, tau2, rho, gamma2, phi,
        digits, transf, targs, verbose=FALSE, control, ...)

## S3 method for class 'rma.uni.selmodel'
confint(object, parm, level, fixed=FALSE, tau2, delta,
        digits, transf, targs, verbose=FALSE, control, ...)

## S3 method for class 'rma.ls'
confint(object, parm, level, fixed=FALSE, alpha,
        digits, transf, targs, verbose=FALSE, control, ...)
```

Arguments

`object`	an object of class `"rma.uni"`, `"rma.mh"`, `"rma.peto"`, `"rma.mv"`, `"rma.uni.selmodel"`, or `"rma.ls"`. The method is not yet implemented for objects of class `"rma.glmm"`.
`parm`	this argument is here for compatibility with the generic function `confint`, but is (currently) ignored.
`fixed`	logical to specify whether confidence intervals for the model coefficients should be returned.
`random`	logical to specify whether a confidence interval for the amount of (residual) heterogeneity should be returned.

`type` optional character string to specify the method to use for computing the confidence interval for the amount of (residual) heterogeneity (either `"QP"`, `"GENQ"`, or `"PL"`).

`sigma2` integer to specify for which σ^2 parameter a confidence interval should be obtained.

`tau2` integer to specify for which τ^2 parameter a confidence interval should be obtained.

`rho` integer to specify for which ρ parameter the confidence interval should be obtained.

`gamma2` integer to specify for which γ^2 parameter a confidence interval should be obtained.

`phi` integer to specify for which ϕ parameter a confidence interval should be obtained.

`delta` integer to specify for which δ parameter a confidence interval should be obtained.

`alpha` integer to specify for which α parameter a confidence interval should be obtained.

`level` numeric value between 0 and 100 to specify the confidence interval level. If unspecified, the default is to take the value from the object.

`digits` integer to specify the number of decimal places to which the results should be rounded. If unspecified, the default is to take the value from the object.

`transf` optional argument to specify a function that should be used to transform the model coefficients and interval bounds (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified under `transf`.

`verbose` logical to specify whether output should be generated on the progress of the iterative algorithms used to obtain the confidence intervals (the default is `FALSE`). See ‘Details’.

`control` list of control values for the iterative algorithms. If unspecified, default values are defined inside the function. See ‘Note’.

`...` other arguments.

Details

Confidence intervals for the model coefficients can be obtained by setting `fixed=TRUE` and are simply the usual Wald-type intervals (which are also shown when printing the fitted object).

Other parameter(s) for which confidence intervals can be obtained depend on the model object:

- For objects of class `"rma.uni"` obtained with the `rma.uni` function, a confidence interval for the amount of (residual) heterogeneity (i.e., τ^2) can be obtained by setting `random=TRUE` (which is the default). The interval is obtained iteratively either via the Q-profile method or via the generalized Q-statistic method (Hartung and Knapp, 2005; Viechtbauer, 2007; Jackson, 2013; Jackson et al., 2014). The latter is automatically used when the model was fitted with `method="GENQ"`, the former is used in all other cases. Either method provides an exact confidence interval for τ^2 in random- and mixed-effects models. The square root of the interval bounds is also returned for easier interpretation. Confidence intervals for I^2 and H^2 are also provided (Higgins & Thompson, 2002). Since I^2 and H^2 are just monotonic transformations of τ^2 (for details, see `print.rma.uni`), the confidence intervals for I^2 and H^2 are also exact. One can also set `type="PL"` to obtain a profile likelihood confidence interval for τ^2 (and corresponding CIs for I^2 and H^2), which would be more consistent with the use of ML/REML estimation, but is not exact (see 'Note').

- For objects of class `"rma.mv"` obtained with the `rma.mv` function, confidence intervals are obtained by default for all (non-fixed) variance and correlation components of the model. Alternatively, one can use the `sigma2`, `tau2`, `rho`, `gamma2`, or `phi` arguments to specify for which variance/correlation parameter a confidence interval should be obtained. Only one of these arguments can be used at a time. A single integer is used to specify the number of the parameter. The function provides profile likelihood confidence intervals for these parameters. It is a good idea to examine the corresponding profile likelihood plots (via the `profile.rma.mv` function) to make sure that the bounds obtained are sensible.
- For selection model objects of class `"rma.uni.selmodel"` obtained with the `selmodel.rma.uni` function, confidence intervals are obtained by default for τ^2 (for models where this is an estimated parameter) and all (non-fixed) selection model parameters. Alternatively, one can choose to obtain a confidence interval only for τ^2 by setting `tau2=TRUE` or for one of the selection model parameters by specifying its number via the `delta` argument. The function provides profile likelihood confidence intervals for these parameters. It is a good idea to examine the corresponding profile likelihood plots (via the `profile.rma.uni.selmodel` function) to make sure that the bounds obtained are sensible.
- For location-scale model objects of class `"rma.ls"` obtained with the `rma.uni` function, confidence intervals are obtained by default for all (non-fixed) scale parameters. Alternatively, one can choose to obtain a confidence interval for one of the scale parameters by specifying its number via the `alpha` argument. The function provides profile likelihood confidence intervals for these parameters. It is a good idea to examine the corresponding profile likelihood plots (via the `profile.rma.ls` function) to make sure that the bounds obtained are sensible.

The methods used to find confidence intervals for these parameters are iterative and require the use of the `uniroot` function. By default, the desired accuracy (`tol`) is set equal to `.Machine$double.eps^0.25` and the maximum number of iterations (`maxiter`) to `1000`. These values can be adjusted with `control=list(tol=value,maxiter=value)`, but the defaults should be adequate for most purposes. If `verbose=TRUE`, output is generated on the progress of the iterative algorithms. This is especially useful when model fitting is slow, in which case finding the confidence interval bounds can also take considerable amounts of time.

When using the `uniroot` function, one must also set appropriate end points of the interval to be searched for the confidence interval bounds. The function tries to set some sensible defaults for the end points, but it may happen that the function is only able to determine that a bound is below/above a certain limit (this is indicated in the output accordingly with < or > signs). It can also happen that the model cannot be fitted or does not converge especially at the extremes of the interval to be searched. This will result in missing (`NA`) bounds and corresponding warnings. It may then be necessary to adjust the end points manually (see 'Note').

Finally, it is also possible that the lower and upper confidence interval bounds for a variance component both fall below zero. Since both bounds then fall outside of the parameter space, the confidence interval then consists of the null/empty set. Alternatively, one could interpret this as a confidence interval with bounds $[0, 0]$ or as indicating 'highly/overly homogeneous' data.

Value

An object of class "confint.rma". The object is a list with either one or two elements (named fixed and random) with the following elements:

estimate	estimate of the model coefficient, variance/correlation component, or selection model parameter.
ci.lb	lower bound of the confidence interval.
ci.ub	upper bound of the confidence interval.

When obtaining confidence intervals for multiple components, the object is a list of class "list.confint.rma", where each element is a "confint.rma" object as described above.

The results are formatted and printed with the print.confint.rma and print.list.confint.rma functions.

Note

When computing a CI for τ^2 for objects of class "rma.uni", the estimate of τ^2 will usually fall within the CI bounds provided by the Q-profile method. However, this is not guaranteed. Depending on the method used to estimate τ^2 and the width of the CI, it can happen that the CI does not actually contain the estimate. Using the empirical Bayes or Paule-Mandel estimator of τ^2 when fitting the model (i.e., using method="EB" or method="PM") usually ensures that the estimate of τ^2 falls within the CI. When method="GENQ" was used to fit the model, the corresponding CI obtained via the generalized Q-statistic method also usually contains the estimate τ^2. When using ML/REML estimation, the profile likelihood CI (obtained when setting type="PL") is guaranteed to contain the estimate of τ^2.

When computing a CI for τ^2 for objects of class "rma.uni", the end points of the interval to be searched for the CI bounds are $[0, 100]$ (or, for the upper bound, ten times the estimate of τ^2, whichever is greater). The upper bound should be large enough for most cases, but can be adjusted with control=list(tau2.max=value). One can also adjust the lower end point with control=list(tau2.min=value). You should only play around with this value if you know what you are doing.

For objects of class "rma.mv", the function provides CIs for the variance/correlation parameters in the model. For variance components, the lower end point of the interval to be searched is set to 0 and the upper end point to the larger of 10 and 100 times the value of the component. For correlations, the function tries to set the lower end point to a sensible default depending on the type of variance structure chosen, while the upper end point is set to 1. One can adjust the lower and/or upper end points with control=list(vc.min=value,vc.max=value). Also, the function tries to adjust the lower/upper end points when the model does not converge at these extremes (the end points are then moved closer to the estimated value of the component). The total number of tries for setting/adjusting the end points in this manner is determined via control=list(eptries=value), with the default being 10 tries.

For objects of class "rma.uni.selmodel" or "rma.ls", the function also sets some sensible defaults for the end points of the interval to be searched for the CI bounds (of the τ^2, δ, and α parameter(s)). One can again adjust the end points and the number of retries (as described above) with control=list(vc.min=value,vc.max=value,eptries=value).

The Q-profile and generalized Q-statistic methods are both exact under the assumptions of the random- and mixed-effects models (i.e., normally distributed observed and true effect sizes or outcomes and known sampling variances). In practice, these assumptions are usually only approximately true, turning CIs for τ^2 also into approximations. Profile likelihood CIs are not exact by construction and rely on the asymptotic behavior of the likelihood ratio statistic, so they may be inaccurate in small samples, but they are inherently consistent with the use of ML/REML estimation.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Hardy, R. J., & Thompson, S. G. (1996). A likelihood approach to meta-analysis with random effects. *Statistics in Medicine*, **15**(6), 619–629. https://doi.org/10.1002/(sici)1097-0258(19960330)15:6%3C619::aid-sim188%3E3.0.co;2-a

Hartung, J., & Knapp, G. (2005). On confidence intervals for the among-group variance in the one-way random effects model with unequal error variances. *Journal of Statistical Planning and Inference*, **127**(1-2), 157–177. https://doi.org/10.1016/j.jspi.2003.09.032

Higgins, J. P. T., & Thompson, S. G. (2002). Quantifying heterogeneity in a meta-analysis. *Statistics in Medicine*, **21**(11), 1539–1558. https://doi.org/10.1002/sim.1186

Jackson, D. (2013). Confidence intervals for the between-study variance in random effects meta-analysis using generalised Cochran heterogeneity statistics. *Research Synthesis Methods*, **4**(3), 220–229. https://doi.org/10.1186/s12874-016-0219-y

Jackson, D., Turner, R., Rhodes, K., & Viechtbauer, W. (2014). Methods for calculating confidence and credible intervals for the residual between-study variance in random effects meta-regression models. *BMC Medical Research Methodology*, **14**, 103. https://doi.org/10.1186/1471-2288-14-103

Viechtbauer, W. (2007). Confidence intervals for the amount of heterogeneity in meta-analysis. *Statistics in Medicine*, **26**(1), 37–52. https://doi.org/10.1002/sim.2514

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mh, rma.peto, rma.glmm, rma.mv, selmodel.rma.uni, profile.rma.uni, profile.rma.mv, profile.rma.uni.selmodel

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### meta-analysis of the log risk ratios using a random-effects model
res <- rma(yi, vi, data=dat, method="REML")

### confidence interval for the total amount of heterogeneity
confint(res)
```

```
### mixed-effects model with absolute latitude in the model
res <- rma(yi, vi, mods = ~ ablat, data=dat)

### confidence interval for the residual amount of heterogeneity
confint(res)

### multilevel random-effects model
res <- rma.mv(yi, vi, random = ~ 1 | district/school, data=dat.konstantopoulos2011)

### profile plots and confidence intervals for the variance components
## Not run:
par(mfrow=c(2,1))
profile(res, sigma2=1, steps=40, cline=TRUE)
sav <- confint(res, sigma2=1)
sav
abline(v=sav$random[1,2:3], lty="dotted")
profile(res, sigma2=2, steps=40, cline=TRUE)
sav <- confint(res, sigma2=2)
sav
abline(v=sav$random[1,2:3], lty="dotted")
## End(Not run)

### multivariate parameterization of the model
res <- rma.mv(yi, vi, random = ~ factor(school) | district, data=dat.konstantopoulos2011)

### profile plots and confidence intervals for the variance component and correlation
## Not run:
par(mfrow=c(2,1))
profile(res, tau2=1, steps=40, cline=TRUE)
sav <- confint(res, tau2=1)
sav
abline(v=sav$random[1,2:3], lty="dotted")
profile(res, rho=1, steps=40, cline=TRUE)
sav <- confint(res, rho=1)
sav
abline(v=sav$random[1,2:3], lty="dotted")
## End(Not run)
```

`contrmat` *Construct Contrast Matrix for Two-Group Comparisons*

Description

The function constructs a matrix that indicates which two groups have been contrasted against each other in each row of a dataset.

Usage

```
contrmat(data, grp1, grp2, last, shorten=FALSE, minlen=2, check=TRUE, append=TRUE)
```

Arguments

`data` a data frame in wide format.

`grp1` either the name (given as a character string) or the position (given as a single number) of the first group variable in the data frame.

`grp2` either the name (given as a character string) or the position (given as a single number) of the second group variable in the data frame.

`last` optional character string to specify which group will be placed in the last column of the matrix (must be one of the groups in the group variables). If not given, the most frequently occurring second group is placed last.

`shorten` logical to specify whether the variable names corresponding to the group names should be shortened (the default is `FALSE`).

`minlen` integer to specify the minimum length of the shortened variable names (the default is 2).

`check` logical to specify whether the variables names should be checked to ensure that they are syntactically valid variable names and if not, they are adjusted (by `make.names`) so that they are (the default is `TRUE`).

`append` logical to specify whether the contrast matrix should be appended to the data frame specified via the `data` argument (the default is `TRUE`). If `append=FALSE`, only the contrast matrix is returned.

Details

The function can be used to construct a matrix that indicates which two groups have been contrasted against each other in each row of a data frame (with `1` for the first group, `-1` for the second group, and `0` otherwise).

The `grp1` and `grp2` arguments are used to specify the group variables in the dataset (either as character strings or as numbers indicating the column positions of these variables in the dataset). Optional argument `last` is used to specify which group will be placed in the last column of the matrix.

If `shorten=TRUE`, the variable names corresponding to the group names are shortened (to at least `minlen`; the actual length might be longer to ensure uniqueness of the variable names).

The examples below illustrate the use of this function.

Value

A matrix with as many variables as there are groups.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

to.wide, dat.senn2013, dat.hasselblad1998

Examples

```
### restructure to wide format
dat <- dat.senn2013
dat <- dat[c(1,4,3,2,5,6)]
dat <- to.wide(dat, study="study", grp="treatment", ref="placebo", grpvars=4:6)
dat

### add contrast matrix
dat <- contrmat(dat, grp1="treatment.1", grp2="treatment.2")
dat

### data in long format
dat <- dat.hasselblad1998
dat

### restructure to wide format
dat <- to.wide(dat, study="study", grp="trt", ref="no_contact", grpvars=6:7)
dat

### add contrast matrix
dat <- contrmat(dat, grp1="trt.1", grp2="trt.2", shorten=TRUE)
dat
```

---

cumul *Cumulative Meta-Analysis for 'rma' Objects*

Description

The functions repeatedly fit the specified model, adding one study at a time to the model.

Usage

```
cumul(x, ...)

## S3 method for class 'rma.uni'
cumul(x, order, digits, transf, targs, progbar=FALSE, ...)
## S3 method for class 'rma.mh'
cumul(x, order, digits, transf, targs, progbar=FALSE, ...)
## S3 method for class 'rma.peto'
cumul(x, order, digits, transf, targs, progbar=FALSE, ...)
```

Arguments

`x` an object of class `"rma.mh"`, `"rma.peto"`, `"rma.uni"`.

`order` optional argument to specify a variable based on which the studies will be ordered for the cumulative meta-analysis.

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.

`transf` optional argument to specify a function that should be used to transform the model coefficients and interval bounds (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified under `transf`.

`progbar` logical to specify whether a progress bar should be shown (the default is `FALSE`).

`...` other arguments.

Details

For `"rma.uni"` objects, the model specified via `x` must be a model without moderators (i.e., either a fixed- or a random-effects model).

If argument `order` is not specified, the studies are added according to their order in the original dataset.

When a variable is specified for `order`, the variable is assumed to be of the same length as the original dataset that was used in the model fitting. Any subsetting and removal of studies with missing values that was applied during the model fitting is also automatically applied to the variable specified via the `order` argument. See ‘Examples’.

Value

An object of class `c("list.rma","cumul.rma")`. The object is a list containing the following components:

`estimate` estimated (average) outcomes.

`se` corresponding standard errors.

`zval` corresponding test statistics.

`pval` corresponding p-values.

`ci.lb` lower bounds of the confidence intervals.

`ci.ub` upper bounds of the confidence intervals.

`Q` test statistics for the test of heterogeneity.

`Qp` corresponding p-values.

`tau2` estimated amount of heterogeneity (only for random-effects models).

`I2` values of I^2.

`H2` values of H^2.

`...` other arguments.

When the model was fitted with `test="t"` or `test="knha"`, then `zval` is called `tval` in the object that is returned by the function.

The object is formatted and printed with `print.list.rma`. A forest plot showing the results from the cumulative meta-analysis can be obtained with `forest.cumul.rma`. Alternatively, `plot.cumul.rma` can also be used to visualize the results.

Note

When using the `transf` option, the transformation is applied to the estimated coefficients and the corresponding interval bounds. The standard errors are then set equal to `NA` and are omitted from the printed output.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Chalmers, T. C., & Lau, J. (1993). Meta-analytic stimulus for changes in clinical trials. *Statistical Methods in Medical Research*, **2**(2), 161–172. https://doi.org/10.1177/096228029300200204

Lau, J., Schmid, C. H., & Chalmers, T. C. (1995). Cumulative meta-analysis of clinical trials builds evidence for exemplary medical care. *Journal of Clinical Epidemiology*, **48**(1), 45–57. https://doi.org/10.1016/0895-4356(94)00106-z

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`forest.cumul.rma`, `plot.cumul.rma`

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit random-effects model
res <- rma(yi, vi, data=dat)

### cumulative meta-analysis (in the order of publication year)
cumul(res, transf=exp, order=dat$year)

### meta-analysis of the (log) risk ratios using the Mantel-Haenszel method
res <- rma.mh(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### cumulative meta-analysis
cumul(res, order=dat.bcg$year)
cumul(res, order=dat.bcg$year, transf=TRUE)

### meta-analysis of the (log) odds ratios using Peto's method
res <- rma.mh(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)
```

```
### cumulative meta-analysis
cumul(res, order=dat.bcg$year)
cumul(res, order=dat.bcg$year, transf=TRUE)

### make first log risk ratio missing and fit model without study 2; then the
### variable specified via 'order' should still be of the same length as the
### original dataset; subsetting and removal of studies with missing values is
### automatically done by the cumul() function
dat$yi[1] <- NA
res <- rma(yi, vi, data=dat, subset=-2)
cumul(res, transf=exp, order=dat$year)
```

`dat.anand1999` *Studies on the Effectiveness of Oral Anticoagulants in Patients with Coronary Artery Disease*

Description

Results from 34 trials examining the effectiveness of oral anticoagulants in patients with coronary artery disease.

Usage

`dat.anand1999`

Format

The data frame contains the following columns:

study	`character`	author(s) or trial name
year	`numeric`	publication year
intensity	`character`	intensity of anticoagulation (low, medium, or high)
asp.t	`numeric`	concomitant use of aspirin in the treatment group (0 = no, 1 = yes)
asp.c	`numeric`	concomitant use of aspirin in the control group (0 = no, 1 = yes)
ai	`numeric`	number of deaths in the treatment group
n1i	`numeric`	number of patients in the treatment group
ci	`numeric`	number of deaths in the control group
n2i	`numeric`	number of patients in the control group

Details

The dataset includes the results from 34 randomized clinical trials that examined the effectiveness of oral anticoagulants in patients with coronary artery disease. The results given here are focused on the total mortality in the treatment versus control groups.

Note

Strictly speaking, there are only 31 trials, since Breddin et al. (1980) and ATACS (1990) are multi-arm trials.

According to a correction, `dat.anand1999$ci[29]` should be 1. But then `dat.anand1999$ci[21]` would also have to be 1 (if these data indeed refer to the same control group). This appears contradictory, so this correction was not made.

Source

Anand, S. S., & Yusuf, S. (1999). Oral anticoagulant therapy in patients with coronary artery disease: A meta-analysis. *Journal of the American Medical Association*, **282**(21), 2058–2067. https://doi.org/10.1001/jama.282.21.2058

Examples

```
### copy data into 'dat' and examine data
dat <- dat.anand1999
dat

### High-Intensity OA vs Control
rma.mh(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat,
       subset=(intensity=="high" & asp.t==0 & asp.c==0), digits=2)

### High- or Moderate-Intensity OA vs Aspirin
rma.mh(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat,
       subset=(intensity %in% c("high","moderate") & asp.t==0 & asp.c==1), digits=2)

### Moderate-Intensity OA vs Control
rma.mh(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat,
       subset=(intensity=="moderate" & asp.t==0 & asp.c==0), digits=2)

### High- or Moderate-Intensity OA and Aspirin vs Aspirin
rma.mh(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat,
       subset=(intensity %in% c("high","moderate") & asp.t==1 & asp.c==1), digits=2)

### Low-Intensity OA and Aspirin vs Aspirin
rma.mh(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat,
       subset=(intensity=="low" & asp.t==1 & asp.c==1), digits=2)
```

`dat.assink2016` *Studies on the Association between Recidivism and Mental Health*

Description

Results from 17 studies on the association between recidivism and mental health in delinquent juveniles.

Usage

```
dat.assink2016
```

Format

The data frame contains the following columns:

study	`numeric`	study id number
esid	`numeric`	effect size within study id number
id	`numeric`	row id number
yi	`numeric`	standardized mean difference
vi	`numeric`	corresponding sampling variance
pubstatus	`numeric`	published study (0 = no; 1 = yes)
year	`numeric`	publication year of the study (approximately mean centered)
deltype	`character`	type of delinquent behavior in which juveniles could have recidivated (either general, overt, or cove

Details

The studies included in this dataset (which is a subset of the data used in Assink et al., 2015) compared the difference in recidivism between delinquent juveniles with a mental health disorder and a comparison group of juveniles without a mental health disorder. Since studies differed in the way recidivism was defined and assessed, results are given in terms of standardized mean differences, with positive values indicating a higher prevalence of recidivism in the group of juveniles with a mental health disorder.

Multiple effect size estimates could be extracted from most studies (e.g., for different delinquent behaviors in which juveniles could have recidivated), necessitating the use of appropriate models/methods for the analysis. Assink and Wibbelink (2016) illustrate the use of multilevel meta-analytic models for this purpose.

Note

The `year` variable is not constant within study 3, as this study refers to two different publications using the same data.

Source

Assink, M., & Wibbelink, C. J. M. (2016). Fitting three-level meta-analytic models in R: A step-by-step tutorial. *The Quantitative Methods for Psychology*, **12**(3), 154–174. https://doi.org/10.20982/tqmp.12.3.p154

References

Assink, M., van der Put, C. E., Hoeve, M., de Vries, S. L. A., Stams, G. J. J. M., & Oort, F. J. (2015). Risk factors for persistent delinquent behavior among juveniles: A meta-analytic review. *Clinical Psychology Review*, **42**, 47–61. https://doi.org/10.1016/j.cpr.2015.08.002

Examples

```
### copy data into 'dat' and examine data
dat <- dat.assink2016
```

```
dat

### fit multilevel model
res <- rma.mv(yi, vi, random = ~ 1 | study/esid, data=dat)
res

### use cluster-robust inference methods
robust(res, cluster=dat$study)

### LRTs for the variance components
res0 <- rma.mv(yi, vi, random = ~ 1 | study/esid, data=dat, sigma2=c(0,NA))
anova(res0, res)
res0 <- rma.mv(yi, vi, random = ~ 1 | study/esid, data=dat, sigma2=c(NA,0))
anova(res0, res)

### examine some potential moderators via meta-regression
rma.mv(yi, vi, mods = ~ pubstatus, random = ~ 1 | study/esid, data=dat)
rma.mv(yi, vi, mods = ~ year, random = ~ 1 | study/esid, data=dat)
dat$deltype <- factor(dat$deltype)
dat$deltype <- relevel(dat$deltype, ref="general")
rma.mv(yi, vi, mods = ~ deltype, random = ~ 1 | study/esid, data=dat)
rma.mv(yi, vi, mods = ~ year + deltype, random = ~ 1 | study/esid, data=dat)
```

dat.bangertdrowns2004 *Studies on the Effectiveness of Writing-to-Learn Interventions*

Description

Results from 48 studies on the effectiveness of school-based writing-to-learn interventions on academic achievement.

Usage

```
dat.bangertdrowns2004
```

Format

The data frame contains the following columns:

id	numeric	study number
author	character	study author(s)
year	numeric	publication year
grade	numeric	grade level (1 = elementary; 2 = middle; 3 = high-school; 4 = college)
length	numeric	treatment length (in weeks)
minutes	numeric	minutes per assignment
wic	numeric	writing tasks were completed in class (0 = no; 1 = yes)
feedback	numeric	feedback on writing was provided (0 = no; 1 = yes)
info	numeric	writing contained informational components (0 = no; 1 = yes)
pers	numeric	writing contained personal components (0 = no; 1 = yes)

imag	`numeric`	writing contained imaginative components (0 = no; 1 = yes)
meta	`numeric`	prompts for metacognitive reflection (0 = no; 1 = yes)
subject	`character`	subject matter
ni	`numeric`	total sample size of the study
yi	`numeric`	standardized mean difference
vi	`numeric`	corresponding sampling variance

Details

In each of the studies included in this meta-analysis, an experimental group (i.e., a group of students that received instruction with increased emphasis on writing tasks) was compared against a control group (i.e., a group of students that received conventional instruction) with respect to some content-related measure of academic achievement (e.g., final grade, an exam/quiz/test score). The outcome measure for this meta-analysis was the standardized mean difference (with positive values indicating a higher mean level of academic achievement in the intervention group).

The standardized mean differences given here are bias-corrected and therefore differ slightly from the values reported in the article. Also, since only the total sample size is given in the article, the sampling variances were computed under the assumption that $n_{i1} = n_{i2} = n_i/2$.

Source

Bangert-Drowns, R. L., Hurley, M. M., & Wilkinson, B. (2004). The effects of school-based writing-to-learn interventions on academic achievement: A meta-analysis. *Review of Educational Research*, **74**(1), 29–58. https://doi.org/10.3102/00346543074001029

Examples

```
### copy data into 'dat' and examine data
dat <- dat.bangertdrowns2004
dat

### fit random-effects model
res <- rma(yi, vi, data=dat)
res
```

`dat.baskerville2012` *Studies on the Effectiveness of Practice Facilitation Interventions*

Description

Results from 23 studies on the effectiveness of practice facilitation interventions within the primary care practice setting.

Usage

```
dat.baskerville2012
```

Format

The data frame contains the following columns:

author	`character`	study author(s)
year	`numeric`	publication year
score	`numeric`	quality score (0 to 12 scale)
design	`character`	study design (cct = controlled clinical trial, rct = randomized clinical trial, crct = cluster randomize
alloconc	`numeric`	allocation concealed (0 = no, 1 = yes)
blind	`numeric`	single- or double-blind study (0 = no, 1 = yes)
itt	`numeric`	intention to treat analysis (0 = no, 1 = yes)
fumonths	`numeric`	follow-up months
retention	`numeric`	retention (in percent)
country	`character`	country where study was conducted
outcomes	`numeric`	number of outcomes assessed
duration	`numeric`	duration of intervention
pperf	`numeric`	practices per facilitator
meetings	`numeric`	(average) number of meetings
hours	`numeric`	(average) hours per meeting
tailor	`numeric`	intervention tailored to the context and needs of the practice (0 = no, 1 = yes)
smd	`numeric`	standardized mean difference
se	`numeric`	corresponding standard error

Details

Baskerville et al. (2012) describe outreach or practice facilitation as a "multifaceted approach that involves skilled individuals who enable others, through a range of intervention components and approaches, to address the challenges in implementing evidence-based care guidelines within the primary care setting". The studies included in this dataset examined the effectiveness of practice facilitation interventions for improving some relevant evidence-based practice behavior. The effect was quantified in terms of a standardized mean difference, comparing the change (from pre- to post-intervention) in the intervention versus the comparison group (or the difference from baseline in prospective cohort studies).

Source

Baskerville, N. B., Liddy, C., & Hogg, W. (2012). Systematic review and meta-analysis of practice facilitation within primary care settings. *Annals of Family Medicine*, **10**(1), 63–74. https://doi.org/10.1370/afm.1312

Examples

```
### copy data into 'dat' and examine data
dat <- dat.baskerville2012
dat

### random-effects model
res <- rma(smd, sei=se, data=dat, method="DL")
print(res, digits=2)

### funnel plot
funnel(res, xlab="Standardized Mean Difference", ylim=c(0,0.6))

### rank and regression tests for funnel plot asymmetry
ranktest(res)
```

```
regtest(res)

### meta-regression analyses examining various potential moderators
rma(smd, sei=se, mods = ~ score, data=dat, method="DL")
rma(smd, sei=se, mods = ~ alloconc, data=dat, method="DL")
rma(smd, sei=se, mods = ~ blind,    data=dat, method="DL")
rma(smd, sei=se, mods = ~ itt,      data=dat, method="DL")
rma(smd, sei=se, mods = ~ duration, data=dat, method="DL")
rma(smd, sei=se, mods = ~ tailor,   data=dat, method="DL")
rma(smd, sei=se, mods = ~ pperf,    data=dat, method="DL")
rma(smd, sei=se, mods = ~ I(meetings * hours), data=dat, method="DL")
```

---

dat.begg1989 *Studies on Bone-Marrow Transplantation versus Chemotherapy for the Treatment of Leukemia*

Description

Results from controlled and uncontrolled studies on the effectiveness of allogeneic bone-marrow transplantation (BMT) and conventional chemotherapy (CMO) in the treatment of acute nonlymphocytic leukemia.

Usage

```
dat.begg1989
```

Format

The data frame contains the following columns:

study	`numeric`	study number
trt	`character`	treatment (BMT or CMO)
arms	`numeric`	number of arms in the study (1 = uncontrolled studies; 2 = controlled studies)
yi	`numeric`	2-year disease-free survival rates
sei	`numeric`	corresponding standard errors
vi	`numeric`	corresponding sampling variances

Details

The dataset includes the results from controlled and uncontrolled studies on the 2-year disease-free survival rate in patients with acute nonlymphocytic leukemia receiving either allogeneic bone-marrow transplantation (BMT) or conventional chemotherapy (CMO). In the controlled (two-arm) studies (studies 1-4), a cohort of patients in complete remission and potentially eligible for BMT was assembled, and those who consented and for whom a donor could be found received BMT, with the remaining patients used as controls (receiving CMO). In the uncontrolled (one-arm) studies (studies 5-16), only a single group was studied, receiving either BMT or CMO.

The data in this dataset were obtained from Table 1 in Begg and Pilote (1991, p. 902).

Source

Begg, C. B., & Pilote, L. (1991). A model for incorporating historical controls into a meta-analysis. *Biometrics*, **47**(3), 899–906. https://doi.org/10.2307/2532647

References

Begg, C. B., Pilote, L., & McGlave, P. B. (1989). Bone marrow transplantation versus chemotherapy in acute non-lymphocytic leukemia: A meta-analytic review. *European Journal of Cancer and Clinical Oncology*, **25**(11), 1519–1523. https://doi.org/10.1016/0277-5379(89)90291-5

Examples

```
### copy data into 'dat' and examine data
dat <- dat.begg1989
dat

### turn trt and arms into factors and set reference levels
dat$trt  <- relevel(factor(dat$trt), ref="CMO")
dat$arms <- relevel(factor(dat$arms), ref="2")

### create data frame with the treatment differences for the controlled studies
dat2 <- data.frame(yi = dat$yi[c(1,3,5,7)] - dat$yi[c(2,4,6,8)],
                   vi = dat$vi[c(1,3,5,7)] + dat$vi[c(2,4,6,8)])
dat2

### DerSimonian and Laird method using the treatment differences
res <- rma(yi, vi, data=dat2, method="DL", digits=2)
res

### Begg & Pilote (1991) model incorporating the uncontrolled studies
res <- rma.mv(yi, vi, mods = ~ trt, random = ~ 1 | study,
              data=dat, method="ML", digits=2)
res

### model involving bias terms for the uncontrolled studies
res <- rma.mv(yi, vi, mods = ~ trt + trt:arms, random = ~ 1 | study,
              data=dat, method="ML", digits=2)
res

### model with a random treatment effect
res <- rma.mv(yi, vi, mods = ~ trt, random = list(~ 1 | study, ~ trt | study),
              struct="UN", tau2=c(0,NA), rho=0, data=dat, method="ML", digits=2)
res

### model with a random treatment effect, but with equal variances in both arms
res <- rma.mv(yi, vi, mods = ~ trt, random = list(~ 1 | study, ~ trt | study),
              struct="CS", rho=0, data=dat, method="ML", digits=2)
res
```

`dat.berkey1998` *Studies on Treatments for Periodontal Disease*

Description

Results from 5 trials comparing surgical and non-surgical treatments for medium-severity periodontal disease one year after treatment.

Usage

```
dat.berkey1998
```

Format

The data frame contains the following columns:

trial	`numeric`	trial number
author	`character`	study author(s)
year	`numeric`	publication year
ni	`numeric`	number of patients
outcome	`character`	outcome (PD = probing depth; AL = attachment level)
yi	`numeric`	observed mean difference in outcome (surgical versus non-surgical)
vi	`numeric`	corresponding sampling variance
v1i	`numeric`	variances and covariances of the observed effects
v2i	`numeric`	variances and covariances of the observed effects

Details

The dataset includes the results from 5 trials that compared surgical and non-surgical methods for the treatment of medium-severity periodontal disease. Reported outcomes include the change in probing depth (PD) and attachment level (AL) one year after the treatment. The outcome measure used for this meta-analysis was the (raw) mean difference, calculated in such a way that positive values indicate that surgery was more effective than non-surgical treatment in decreasing the probing depth and increasing the attachment level (so, the results from the various trials indicate that surgery is preferable for reducing the probing depth, while non-surgical treatment is preferable for increasing the attachment level). Since each trial provides effect size estimates for both outcomes, the estimates are correlated. A multivariate model can be used to meta-analyze the two outcomes simultaneously.

The `v1i` and `v2i` values are the variances and covariances of the observed effects. In particular, for each study, variables `v1i` and `v2i` form a 2×2 variance-covariance matrix of the observed effects, with the diagonal elements corresponding to the sampling variances of the mean differences (the first for probing depth, the second for attachment level) and the off-diagonal value corresponding to the covariance of the two mean differences. Below, the full (block diagonal) variance-covariance for all studies is constructed from these two variables.

Source

Berkey, C. S., Antczak-Bouckoms, A., Hoaglin, D. C., Mosteller, F., & Pihlstrom, B. L. (1995). Multiple-outcomes meta-analysis of treatments for periodontal disease. *Journal of Dental Research*, **74**(4), 1030–1039. https://doi.org/10.1177/00220345950740040201

Berkey, C. S., Hoaglin, D. C., Antczak-Bouckoms, A., Mosteller, F., & Colditz, G. A. (1998). Meta-analysis of multiple outcomes by regression with random effects. *Statistics in Medicine*, **17**(22), 2537–2550. https://doi.org/10.1002/(sici)1097-0258(19981130)17:22<2537::aid-sim953>3.0.co;2-c

Examples

```
### copy data into 'dat' and examine data
dat <- dat.berkey1998
dat

### construct list with the variance-covariance matrices of the observed outcomes for the studies
V <- lapply(split(dat[c("v1i", "v2i")], dat$trial), as.matrix)

### construct block diagonal matrix
V <- bldiag(V)

### fit multiple outcomes (meta-regression) model (with REML estimation)
res <- rma.mv(yi, V, mods = ~ outcome - 1, random = ~ outcome | trial, struct="UN", data=dat)
print(res, digits=3)

### test/estimate difference between the two outcomes
anova(res, X=c(1,-1))

### fit model including publication year as moderator for both outcomes (with ML estimation)
res <- rma.mv(yi, V, mods = ~ outcome + outcome:I(year - 1983) - 1,
              random = ~ outcome | trial, struct="UN", data=dat, method="ML")
print(res, digits=3)
```

dat.bonett2010 *Studies on the Reliability of the CES-D Scale*

Description

Results from 9 studies on the reliability of the Center for Epidemiologic Studies Depression (CES-D) Scale administered to children providing care to an elderly parent.

Usage

```
dat.bonett2010
```

Format

The data frame contains the following columns:

study	`numeric`	study number
source	`character`	source of data
ni	`numeric`	sample size
mi	`numeric`	number of items in the scale
ai	`numeric`	observed value of Cronbach's alpha
caregivers	`character`	gender of the children in the sample

Details

The Center for Epidemiologic Studies Depression (CES-D) Scale is a 20-item questionnaire assessing various symptoms of depression, with each item scored on a 4-point scale. The scale has been used in several studies to examine depressive symptoms in children providing care to an elderly parent. The dataset includes information on the reliability of the scale as measured with Cronbach's alpha in 9 such studies. Also, the gender composition of the children in each sample is indicated.

Source

Bonett, D. G. (2010). Varying coefficient meta-analytic methods for alpha reliability. *Psychological Methods*, **15**(4), 368–385. https://doi.org/10.1037/a0020142

References

Bonett, D. G. (2002). Sample size requirements for testing and estimating coefficient alpha. *Journal of Educational and Behavioral Statistics*, **27**(4), 335–340. https://doi.org/10.3102/10769986027004335

Hakstian, A. R., & Whalen, T. E. (1976). A k-sample significance test for independent alpha coefficients. *Psychometrika*, **41**(2), 219–231. https://doi.org/10.1007/BF02291840

Examples

```
### copy data into 'dat' and examine data
dat <- dat.bonett2010
dat

### meta-analysis using the raw alpha values
res <- rma(measure="ARAW", ai=ai, mi=mi, ni=ni, data=dat)
res

### meta-analysis using transformed alpha values (using the
### transformation suggested by Hakstian & Whalen, 1976)
res <- rma(measure="AHW", ai=ai, mi=mi, ni=ni, data=dat)
res
predict(res, transf=transf.iahw)

### meta-analysis using transformed alpha values (using the
### transformation suggested by Bonett, 2002)
res <- rma(measure="ABT", ai=ai, mi=mi, ni=ni, data=dat)
res
predict(res, transf=transf.iabt)

### forest plot
forest(res, slab=dat$source, header=TRUE, atransf=transf.iabt, refline=coef(res))
```

```
### examine whether female/mixed samples yield different alphas (with raw alphas)
res <- rma(measure="ARAW", ai=ai, mi=mi, ni=ni, mods = ~ caregivers, data=dat)
res
predict(res, newmods=c(0,1), digits=2)
```

---

dat.bornmann2007 *Studies on Gender Differences in Grant and Fellowship Awards*

Description

Results from 21 studies on gender differences in grant and fellowship awards.

Usage

```
dat.bornmann2007
```

Format

The data frame contains the following columns:

study	`character`	study reference
obs	`numeric`	observation within study
doctype	`character`	document type
gender	`character`	gender of the study authors
year	`numeric`	(average) cohort year
org	`character`	funding organization / program
country	`character`	country of the funding organization / program
type	`character`	fellowship or grant application
discipline	`character`	discipline / field
waward	`numeric`	number of women who received a grant/fellowship award
wtotal	`numeric`	number of women who applied for an award
maward	`numeric`	number of men who received a grant/fellowship award
mtotal	`numeric`	number of men who applied for an award

Details

The studies in this dataset examine whether the chances of receiving a grant or fellowship award differs for men and women. Note that many studies provide multiple comparisons (e.g., for different years / cohorts / disciplines). A multilevel meta-analytic model can be used to account for the multilevel structure in these data.

Source

Bornmann, L., Mutz, R., & Daniel, H. (2007). Gender differences in grant peer review: A meta-analysis. *Journal of Informetrics*, **1**(3), 226–238. https://doi.org/10.1016/j.joi.2007.03.001

References

Marsh, H. W., Bornmann, L., Mutz, R., Daniel, H.-D., & O'Mara, A. (2009). Gender effects in the peer reviews of grant proposals: A comprehensive meta-analysis comparing traditional and multilevel approaches. *Review of Educational Research*, **79**(3), 1290–1326. https://doi.org/10.3102/0034654309334143

Examples

```
### copy data into 'dat' and examine data
dat <- dat.bornmann2007
dat

### calculate log odds ratios and corresponding sampling variances
dat <- escalc(measure="OR", ai=waward, n1i=wtotal, ci=maward, n2i=mtotal, data=dat)

### fit multilevel meta-analytic model
res <- rma.mv(yi, vi, random = ~ 1 | study/obs, data=dat)
res

### estimated average odds ratio (with 95% CI/PI)
predict(res, transf=exp, digits=2)

### test for a difference between fellowship and grant applications
res <- rma.mv(yi, vi, mods = ~ type, random = ~ 1 | study/obs, data=dat)
res
predict(res, newmods=0:1, transf=exp, digits=2)
```

`dat.bourassa1996` *Studies on the Association between Handedness and Eye-Dominance*

Description

Results from 47 studies on the association between handedness and eye-dominance.

Usage

```
dat.bourassa1996
```

Format

The data frame contains the following columns:

study	`numeric`	study number
sample	`numeric`	sample number
author	`character`	(first) author
year	`numeric`	publication year
selection	`character`	selection of subjects on the basis of eyedness or handedness
investigator	`character`	investigator (psychologist, educationalist, or other)
hand_assess	`character`	method to assess handedness (questionnaire or performance based)

eye_assess	`character`	method to assess eyedness (see 'Details')
mage	`numeric`	mean age of sample
lh.le	`numeric`	number of left-handed left-eyed individuals
lh.re	`numeric`	number of left-handed right-eyed individuals
rh.le	`numeric`	number of right-handed left-eyed individuals
rh.re	`numeric`	number of right-handed right-eyed individuals
sex	`character`	sex of the sample (combined, male, or female)

Details

The 47 studies included in this meta-analysis examined the association between handedness and eye-dominance (ocular dominance or eyedness). Results are given in terms of 2×2 tables, indicating the number of left-handed left-eyed, left-handed right-eyed, right-handed left-eyed, and right-handed right-eyed individuals. Note that some studies included multiple (independent) samples, so that the meta-analysis included 54 samples in total. Also, for some studies, the combined data of the males and females are further broken down into the two subgroups.

In some studies, there was indication that the selection of subjects was not random with respect to handedness and/or eyedness. While this should not influence the size of the association as measured with the odds ratio, this invalidates those studies for assessing the overall percentage of left-eyed and left-handed individuals.

Handedness was assessed in the individual studies either based on a questionnaire or inventory or based on task performance. Eyedness was assessed based on various methods: `E.1` methods are based on task performance, while `E.2.a` denotes assessment based on a questionnaire. The performance based methods could be further broken down into: `E.1.a.i` (monocular procedure with object/instrument held in one hand), `E.1.a.ii` (monocular procedure with object/instrument held in both hands), `E.1.b` (binocular procedure), `E.1.c` (a combination of the previous methods), and `E.1.d` (some other method).

Source

Bourassa, D. C., McManus, I. C., & Bryden, M. P. (1996). Handedness and eye-dominance: A meta-analysis of their relationship. *Laterality*, **1**(1), 5–34. https://doi.org/10.1080/713754206

Examples

```
### copy data into 'dat'
dat <- dat.bourassa1996

### calculate log(OR) and corresponding sampling variance with 1/2 correction
dat <- escalc(measure="OR", ai=lh.le, bi=lh.re, ci=rh.le, di=rh.re, data=dat, add=1/2, to="all")
dat

### overall association between handedness and eyedness
res <- rma(yi, vi, data=dat, subset=sex=="combined")
res
predict(res, transf=exp, digits=2)
```

`dat.cannon2006` *Studies on the Effectiveness of Intensive Versus Moderate Statin Therapy for Preventing Coronary Death or Myocardial Infarction*

Description

Results from 4 trials examining the effectiveness of intensive (high dose) versus moderate (standard dose) statin therapy for preventing coronary death or myocardial infarction.

Usage

```
dat.cannon2006
```

Format

The data frame contains the following columns:

trial	`character`	trial name
pop	`character`	study population (post-ACS: post acute coronary syndrome; stable CAD: stable coronary artery disease)
nt	`numeric`	number of patients in the high dose group
nc	`numeric`	number of patients in the standard dose group
ep1t	`numeric`	number of events in the high dose group for end point 1: coronary death or non-fatal myocardial infarcti
ep1c	`numeric`	number of events in the standard dose group for end point 1: coronary death or non-fatal myocardial infa
ep2t	`numeric`	number of events in the high dose group for end point 2: coronary death or any cardiovascular event (MI
ep2c	`numeric`	number of events in the standard dose group for end point 2: coronary death or any cardiovascular event
ep3t	`numeric`	number of events in the high dose group for end point 3: cardiovascular death
ep3c	`numeric`	number of events in the standard dose group for end point 3: cardiovascular death
ep4t	`numeric`	number of events in the high dose group for end point 4: non-cardiovascular death
ep4c	`numeric`	number of events in the standard dose group for end point 4: non-cardiovascular death
ep5t	`numeric`	number of events in the high dose group for end point 5: deaths (all-cause mortality)
ep5c	`numeric`	number of events in the standard dose group for end point 5: deaths (all-cause mortality)
ep6t	`numeric`	number of events in the high dose group for end point 6: stroke
ep6c	`numeric`	number of events in the standard dose group for end point 6: stroke

Details

The data were obtained from Figures 2, 3, 4, and 5 in Cannon et al. (2006). The authors used the Mantel-Haenszel method for combining the results from the 4 trials. This approach is implemented in the `rma.mh` function.

Source

Cannon, C. P., Steinberg, B. A., Murphy, S. A., Mega, J. L., & Braunwald, E. (2006). Meta-analysis of cardiovascular outcomes trials comparing intensive versus moderate statin therapy. *Journal of the American College of Cardiology*, **48**(3), 438–445. https://doi.org/10.1016/j.jacc.2006.04.070

Examples

```
### copy data into 'dat' and examine data
dat <- dat.cannon2006
dat

### meta-analysis of log odds ratios using the MH method for endpoint 1
res <- rma.mh(measure="OR", ai=ep1t, n1i=nt, ci=ep1c, n2i=nc, data=dat, slab=trial)
print(res, digits=2)

### forest plot
forest(res, xlim=c(-.8,.8), atransf=exp, at=log(c(2/3, 1, 3/2)),
       header=TRUE, top=2, cex=1.2, xlab="Odds Ratio")
mtext("(high dose better)", side=1, line=par("mgp")[1]-0.5, at=log(2/3), cex=1.2, font=3)
mtext("(standard dose better)", side=1, line=par("mgp")[1]-0.5, at=log(3/2), cex=1.2, font=3)
```

dat.cohen1981 *Studies on the Relationship between Course Instructor Ratings and Student Achievement*

Description

Results from 20 studies on the correlation between course instructor ratings and student achievement.

Usage

```
dat.cohen1981
```

Format

The data frame contains the following columns:

study	`character`	study author(s) and year
sample	`character`	course type
control	`character`	ability control
ni	`numeric`	sample size of the study (number of sections)
ri	`numeric`	observed correlation

Details

The studies included in this dataset examined to what extent students' ratings of a course instructor correlated with their achievement in the course. Instead of correlating individual ratings and achievement scores, the studies were carried out in multisection courses, in which the sections had different instructors but all sections used a common achievement measure (e.g., a final exam). The correlation coefficients reflect the correlation between the mean instructor rating and the mean achievement score of each section. Hence, the unit of analysis are the sections, not the individuals.

Note that this dataset (extracted from Table A.3 in Cooper & Hedges, 1994) only contains studies with at least 10 sections.

Source

Cooper, H., & Hedges, L. V. (1994). Appendix A: Data Sets. In H. Cooper & L. V. Hedges (Eds.), *The handbook of research synthesis* (pp. 543-547). New York: Russell Sage Foundation.

References

Cohen, P. A. (1981). Student ratings of instruction and student achievement: A meta-analysis of multisection validity studies. *Review of Educational Research*, **51**(3), 281–309. https://doi.org/10.3102/00346543051003281

Examples

```
### copy data into 'dat'
dat <- dat.cohen1981

### calculate r-to-z transformed correlations and corresponding sampling variances
dat <- escalc(measure="ZCOR", ri=ri, ni=ni, data=dat[c(1,4,5)])
dat

### meta-analysis of the transformed correlations using a random-effects model
res <- rma(yi, vi, data=dat, digits=2)
res

### predicted average correlation with 95% CI
predict(res, transf=transf.ztor)
```

dat.colditz1994 *Studies on the Effectiveness of the BCG Vaccine Against Tuberculosis*

Description

Results from 13 studies examining the effectiveness of the Bacillus Calmette-Guerin (BCG) vaccine against tuberculosis.

Usage

```
dat.colditz1994
dat.bcg
```

Format

The data frame contains the following columns:

trial	`numeric`	trial number
author	`character`	author(s)
year	`numeric`	publication year
tpos	`numeric`	number of TB positive cases in the treated (vaccinated) group
tneg	`numeric`	number of TB negative cases in the treated (vaccinated) group
cpos	`numeric`	number of TB positive cases in the control (non-vaccinated) group

cneg	`numeric`	number of TB negative cases in the control (non-vaccinated) group
ablat	`numeric`	absolute latitude of the study location (in degrees)
alloc	`character`	method of treatment allocation (random, alternate, or systematic assignment)

Details

The 13 studies provide data in terms of 2×2 tables in the form:

	TB positive	TB negative
vaccinated group	`tpos`	`tneg`
control group	`cpos`	`cneg`

The goal of the meta-analysis was to examine the overall effectiveness of the BCG vaccine for preventing tuberculosis and to examine moderators that may potentially influence the size of the effect.

The dataset has been used in several publications to illustrate meta-analytic methods (see 'References').

Source

Colditz, G. A., Brewer, T. F., Berkey, C. S., Wilson, M. E., Burdick, E., Fineberg, H. V., & Mosteller, F. (1994). Efficacy of BCG vaccine in the prevention of tuberculosis: Meta-analysis of the published literature. *Journal of the American Medical Association*, **271**(9), 698–702. https://doi.org/10.1001/jama.1994.035103300760

References

Berkey, C. S., Hoaglin, D. C., Mosteller, F., & Colditz, G. A. (1995). A random-effects regression model for meta-analysis. *Statistics in Medicine*, **14**(4), 395–411. https://doi.org/10.1002/sim.4780140406

van Houwelingen, H. C., Arends, L. R., & Stijnen, T. (2002). Advanced methods in meta-analysis: Multivariate approach and meta-regression. *Statistics in Medicine*, **21**(4), 589–624. https://doi.org/10.1002/sim.1040

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Examples

```
### copy data into 'dat'
dat <- dat.bcg

### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat)
dat

### random-effects model
res <- rma(yi, vi, data=dat)
res

### average risk ratio with 95% CI
predict(res, transf=exp)
```

```
### mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)
res

### predicted average risk ratios for 10-60 degrees absolute latitude
### holding the publication year constant at 1970
predict(res, newmods=cbind(seq(from=10, to=60, by=10), 1970), transf=exp)

### note: the interpretation of the results is difficult because absolute
### latitude and publication year are strongly correlated (the more recent
### studies were conducted closer to the equator)
plot(dat$year, dat$ablat, pch=19, xlab="Publication Year", ylab="Absolute Lattitude")
cor(dat$year, dat$ablat)
```

`dat.collins1985a` *Studies on the Treatment of Upper Gastrointestinal Bleeding by a Histamine H2 Antagonist*

Description

Results from studies examining the effectiveness of histamine H2 antagonists (cimetidine or ranitidine) in treating patients with acute upper gastrointestinal hemorrhage.

Usage

```
dat.collins1985a
```

Format

The data frame contains the following columns:

id	`numeric`	study number
trial	`character`	first author of trial
year	`numeric`	year of publication
ref	`numeric`	reference number
trt	`character`	C = cimetidine, R = ranitidine
ctrl	`character`	P = placebo, AA = antacids, UT = usual treatment
nti	`numeric`	number of patients in treatment group
b.xti	`numeric`	number of patients in treatment group with persistent or recurrent bleedings
o.xti	`numeric`	number of patients in treatment group in need of operation
d.xti	`numeric`	number of patients in treatment group that died
nci	`numeric`	number of patients in control group
b.xci	`numeric`	number of patients in control group with persistent or recurrent bleedings
o.xci	`numeric`	number of patients in control group in need of operation
d.xci	`numeric`	number of patients in control group that died

Details

The data were obtained from Tables 1 and 2 in Collins and Langman (1985). The authors used Peto's (one-step) method for meta-analyzing the 27 trials. This approach is implemented in the rma.peto function. Using the same dataset, van Houwelingen, Zwinderman, and Stijnen (1993) describe some alternative approaches for analyzing these data, including fixed and random-effects conditional logistic models. Those are implemented in the rma.glmm function.

Source

Collins, R., & Langman, M. (1985). Treatment with histamine H2 antagonists in acute upper gastrointestinal hemorrhage. *New England Journal of Medicine*, **313**(11), 660–666. https://doi.org/10.1056/NEJM19850912313

References

van Houwelingen, H. C., Zwinderman, K. H., & Stijnen, T. (1993). A bivariate approach to meta-analysis. *Statistics in Medicine*, **12**(24), 2273–2284. https://doi.org/10.1002/sim.4780122405

Examples

```
### copy data into 'dat' and examine data
dat <- dat.collins1985a
dat

### meta-analysis of log ORs using Peto's method (outcome: persistent or recurrent bleedings)
res <- rma.peto(ai=b.xti, n1i=nti, ci=b.xci, n2i=nci, data=dat)
print(res, digits=2)

## Not run:
### meta-analysis of log ORs using a conditional logistic regression model (FE model)
res <- rma.glmm(measure="OR", ai=b.xti, n1i=nti, ci=b.xci, n2i=nci, data=dat,
                model="CM.EL", method="FE")
summary(res)
predict(res, transf=exp, digits=2)

### plot the likelihoods of the odds ratios
llplot(measure="OR", ai=b.xti, n1i=nti, ci=b.xci, n2i=nci, data=dat,
       lwd=1, refline=NA, xlim=c(-4,4), drop00=FALSE)

### meta-analysis of log odds ratios using a conditional logistic regression model (RE model)
res <- rma.glmm(measure="OR", ai=b.xti, n1i=nti, ci=b.xci, n2i=nci, data=dat,
                model="CM.EL", method="ML")
summary(res)
predict(res, transf=exp, digits=2)

## End(Not run)

### meta-analysis of log ORs using Peto's method (outcome: need for surgery)
res <- rma.peto(ai=o.xti, n1i=nti, ci=o.xci, n2i=nci, data=dat)
print(res, digits=2)

### meta-analysis of log ORs using Peto's method (outcome: death)
```

```
res <- rma.peto(ai=d.xti, n1i=nti, ci=d.xci, n2i=nci, data=dat)
print(res, digits=2)
```

---

dat.collins1985b *Studies on the Effects of Diuretics in Pregnancy*

Description

Results from 9 studies examining the effects of diuretics in pregnancy on various outcomes.

Usage

```
dat.collins1985b
```

Format

The data frame contains the following columns:

id	`numeric`	study number
author	`character`	study author(s)
year	`numeric`	publication year
pre.nti	`numeric`	number of women in treatment group followed up for pre-eclampsia outcome
pre.nci	`numeric`	number of women in control/placebo group followed up for pre-eclampsia outcome
pre.xti	`numeric`	number of women in treatment group with any form of pre-eclampsia
pre.xci	`numeric`	number of women in control/placebo group with any form of pre-eclampsia
oedema	`numeric`	dummy variable indicating whether oedema was a diagnostic criterion
fup.nti	`numeric`	number of women in treatment group followed up for mortality outcomes
fup.nci	`numeric`	number of women in control/placebo group followed up for mortality outcomes
ped.xti	`numeric`	number of perinatal deaths in treatment group
ped.xci	`numeric`	number of perinatal deaths in control/placebo group
stb.xti	`numeric`	number of stillbirths in treatment group
stb.xci	`numeric`	number of stillbirths in control/placebo group
ned.xti	`numeric`	number of neonatal deaths in treatment group
ned.xci	`numeric`	number of neonatal deaths in control/placebo group

Details

The 9 studies in this dataset examined the effects of diuretics in pregnancy on various outcomes, including the presence of any form of pre-eclampsia, perinatal death, stillbirth, and neonatal death.

Source

Collins, R., Yusuf, S., & Peto, R. (1985). Overview of randomised trials of diuretics in pregnancy. *British Medical Journal*, **290**(6461), 17–23. https://doi.org/10.1136/bmj.290.6461.17

Examples

```
### copy data into 'dat'
```

```
dat <- dat.collins1985b

### calculate (log) odds ratio and sampling variance
dat <- escalc(measure="OR", n1i=pre.nti, n2i=pre.nci, ai=pre.xti, ci=pre.xci, data=dat)
summary(dat, digits=2, transf=exp)

### meta-analysis using Peto's method for any form of pre-eclampsia
rma.peto(n1i=pre.nti, n2i=pre.nci, ai=pre.xti, ci=pre.xci, data=dat, digits=2)

### meta-analysis including only studies where oedema was not a diagnostic criterion
rma.peto(n1i=pre.nti, n2i=pre.nci, ai=pre.xti, ci=pre.xci, data=dat, digits=2, subset=(oedema==0))

### meta-analyses of mortality outcomes (perinatal deaths, stillbirths, and neonatal deaths)
rma.peto(n1i=fup.nti, n2i=fup.nci, ai=ped.xti, ci=ped.xci, data=dat, digits=2)
rma.peto(n1i=fup.nti, n2i=fup.nci, ai=stb.xti, ci=stb.xci, data=dat, digits=2)
rma.peto(n1i=fup.nti, n2i=fup.nci, ai=ned.xti, ci=ned.xci, data=dat, digits=2)
```

`dat.craft2003` *Studies on the Relationship between the Competitive State Anxiety Inventory-2 and Sport Performance*

Description

Results from 10 studies on the relationship between the Competitive State Anxiety Inventory-2 (CSAI-2) and sport performance.

Usage

```
dat.craft2003
```

Format

The data frame contains the following columns:

study	`numeric`	study number
ni	`numeric`	sample size
sport	`character`	type of sport (T = team sport, I = individual sport)
ri	`numeric`	correlation coefficient
var1	`character`	variable 1 of the correlation coefficient (see 'Details')
var2	`character`	variable 2 of the correlation coefficient (see 'Details')

Details

The 10 studies included in this dataset are a subset of the studies included in the meta-analysis by Craft et al. (2003) on the relationship between the Competitive State Anxiety Inventory-2 (CSAI-2) and sport performance.

The CSAI-2 has three subscales: cognitive anxiety (`acog`), somatic anxiety (`asom`), and self-confidence (`conf`). The studies included in this dataset administered the CSAI-2 prior to some sport competition and then measured sport performance based on the competition. Most studies provided all 6 correlations (3 for the correlations among the 3 subscales and 3 for the correlations between the subscales and sport performance), but 2 studies (with study numbers 6 and 17) only provided a subset.

Source

Becker, B. J., & Aloe, A. M. (2019). Model-based meta-analysis and related approaches. In H. Cooper, L. V. Hedges, & J. C. Valentine (Eds.), *The handbook of research synthesis and meta-analysis* (3nd ed., pp. 339–363). New York: Russell Sage Foundation.

References

Craft, L. L., Magyar, T. M., Becker, B. J., & Feltz, D. L. (2003). The relationship between the Competitive State Anxiety Inventory-2 and sport performance: A meta-analysis. *Journal of Sport and Exercise Psychology*, **25**(1), 44–65. https://doi.org/10.1123/jsep.25.1.44

Examples

```
### copy data into 'dat'
dat <- dat.craft2003
dat

### construct dataset and var-cov matrix of the correlations
tmp <- rcalc(ri ~ var1 + var2 | study, ni=ni, data=dat)
V <- tmp$V
dat <- tmp$dat

### examine data for study 1
dat[dat$study == 1,]
V[dat$study == 1, dat$study == 1]

### examine data for study 6
dat[dat$study == 6,]
V[dat$study == 6, dat$study == 6]

### examine data for study 17
dat[dat$study == 17,]
V[dat$study == 17, dat$study == 17]

### multivariate random-effects model
res <- rma.mv(yi, V, mods = ~ var1.var2 - 1, random = ~ var1.var2 | study, struct="UN", data=dat)
res
```

---

dat.crede2010 *Studies on the Relationship between Class Attendance and Grades in College Students*

Description

Results from 68 studies on the relationship between class attendence and class performance and/or grade point average in college students.

Usage

```
dat.crede2010
```

Format

The data frame contains the following columns:

studyid	`numeric`	study number
year	`numeric`	publication year
source	`character`	study source (journal, dissertation, other)
sampleid	`numeric`	sample within study number
criterion	`character`	criterion variable (grade, gpa)
class	`character`	class type (science, nonscience)
ni	`numeric`	sample size
ri	`numeric`	observed correlation

Details

The 68 studies included in this dataset provide information about the relationship between class attendance of college students and their performance (i.e., grade) in the class and/or their overall grade point average. Some studies included multiple samples and hence the dataset actually contains 97 correlation coefficients.

The dataset was obtained via personal communication. Note that this dataset differs just slightly from the one used by Credé et al. (2010).

Source

Personal communication.

References

Credé, M., Roch, S. G., & Kieszczynka, U. M. (2010). Class attendance in college: A meta-analytic review of the relationship of class attendance with grades and student characteristics. *Review of Educational Research*, **80**(2), 272–295. https://doi.org/10.3102/0034654310362998

Examples

```
### copy data into 'dat'
dat <- dat.crede2010

### calculate r-to-z transformed correlations and corresponding sampling variances
dat <- escalc(measure="ZCOR", ri=ri, ni=ni, data=dat)

############################################################################
```

```
### meta-analysis for the relationship between attendance and grades
res <- rma(yi, vi, data=dat, subset=criterion=="grade")
res

### estimated average correlation with 95% CI/PI
predict(res, transf=transf.ztor, digits=2)

### examine if relationship between attendance and grades differs for nonscience/science classes
res <- rma(yi, vi, mods = ~ class, data=dat, subset=criterion=="grade")
res

### estimated average correlations for nonscience and science classes
predict(res, newmods=c(0,1), transf=transf.ztor, digits=2)

### examine if relationship between attendance and grades has changed over time
res <- rma(yi, vi, mods = ~ year, data=dat, subset=criterion=="grade")
res

############################################################################

### meta-analysis for the relationship between attendance and GPA
res <- rma(yi, vi, data=dat, subset=criterion=="gpa")
res

### estimated average correlation with 95% CI/PI
predict(res, transf=transf.ztor, digits=2)

### examine if relationship between attendance and GPA has changed over time
res <- rma(yi, vi, mods = ~ year, data=dat, subset=criterion=="gpa")
res

############################################################################

### use a multilevel model to examine the relationship between attendance and grades
res <- rma.mv(yi, vi, random = ~ 1 | studyid/sampleid, data=dat, subset=criterion=="grade")
res
predict(res, transf=transf.ztor, digits=2)

### use a multilevel model to examine the relationship between attendance and gpa
res <- rma.mv(yi, vi, random = ~ 1 | studyid/sampleid, data=dat, subset=criterion=="gpa")
res
predict(res, transf=transf.ztor, digits=2)
```

dat.curtis1998 *Studies on the Effects of Elevated CO2 Levels on Woody Plant Mass*

Description

Results from studies examining the effects of elevated CO2 levels on woody plant mass.

Usage

```
dat.curtis1998
```

Format

The data frame contains the following columns:

id	`numeric`	observation number
paper	`numeric`	paper number
genus	`character`	genus name
species	`character`	species name
fungrp	`character`	plant functional group
co2.ambi	`numeric`	ambient CO2 level (control group)
co2.elev	`numeric`	elevated CO2 level (treatment group)
units	`character`	units for CO2 exposure levels
time	`numeric`	maximum length of time (days) of CO2 exposure
pot	`character`	growing method (see 'Details')
method	`character`	CO2 exposure facility (see 'Details')
stock	`character`	planting stock code
xtrt	`character`	interacting treatment code (see 'Details')
level	`character`	interacting treatment level codes (see 'Details')
m1i	`numeric`	mean plant mass under elevated CO2 level (treatment group)
sd1i	`numeric`	standard deviation of plant mass underelevated CO2 level (treatment group)
n1i	`numeric`	number of observations under elevated CO2 level (treatment group)
m2i	`numeric`	mean plant mass under ambient CO2 level (control group)
sd2i	`numeric`	standard deviation of plant mass under ambient CO2 level (control group)
n2i	`numeric`	number of observations under ambient CO2 level (control group)

Details

The studies included in this dataset compared the total above- plus below-ground biomass (in grams) for plants that were either exposed to ambient (around 35 Pa) and elevated CO2 levels (around twice the ambient level). The `co2.ambi` and `co2.elev` variables indicate the CO2 levels in the control and treatment groups, respectively (with the `units` variable specifying the units for the CO2 exposure levels). Many of the studies also varied one or more additional environmental variables (defined by the `xtrt` and `level` variables):

- NONE = no additional treatment factor
- FERT = soil fertility (either a `CONTROL`, `HIGH`, or `LOW` level)
- LIGHT = light treatment (always a `LOW` light level)
- FERT+L = soil fertility and light (a `LOW` light and soil fertility level)
- H2O = well watered vs drought (either a `WW` or `DRT` level)
- TEMP = temperature treatment (either a `HIGH` or `LOW` level)
- OZONE = ozone exposure (either a `HIGH` or `LOW` level)
- UVB = ultraviolet-B radiation exposure (either a `HIGH` or `LOW` level)

In addition, the studies differed with respect to various design variables, including CO2 exposure duration (time), growing method (pot: number = pot size in liters; GRND = plants rooted in ground; HYDRO = solution or aeroponic culture), CO2 exposure facility (method: GC = growth chamber; GH = greenhouse; OTC = field-based open-top chamber), and planting stock (stock: SEED = plants started from seeds; SAP = plants started from cuttings). The goal of the meta-analysis was to examine the effects of elevated CO2 levels on plant physiology and growth and the interacting effects of the environmental (and design) variables.

Source

Hedges, L. V., Gurevitch, J., & Curtis, P. S. (1999). The meta-analysis of response ratios in experimental ecology. *Ecology*, **80**(4), 1150–1156. https://doi.org/10.1890/0012-9658(1999)080[1150:TMAORR]2.0.CO;2 (data obtained from *Ecological Archives*, E080-008-S1, at: https://esapubs.org/archive/ecol/E080/008/)

References

Curtis, P. S., & Wang, X. (1998). A meta-analysis of elevated CO2 effects on woody plant mass, form, and physiology. *Oecologia*, **113**(3), 299–313. https://doi.org/10.1007/s004420050381

Examples

```
### copy data into 'dat'
dat <- dat.curtis1998

### calculate (log transformed) ratios of means and corresponding sampling variances
dat <- escalc(measure="ROM", m1i=m1i, sd1i=sd1i, n1i=n1i,
                             m2i=m2i, sd2i=sd2i, n2i=n2i, data=dat)
dat

### meta-analysis using a random-effects model
res <- rma(yi, vi, method="DL", data=dat)
res

### average ratio of means with 95% CI
predict(res, transf=exp, digits=2)

### meta-analysis for plants grown under nutrient stress
res <- rma(yi, vi, method="DL", data=dat, subset=(xtrt=="FERT" & level=="LOW"))
predict(res, transf=exp, digits=2)

### meta-analysis for plants grown under low light conditions
res <- rma(yi, vi, method="DL", data=dat, subset=(xtrt=="LIGHT" & level=="LOW"))
predict(res, transf=exp, digits=2)
```

---

dat.dagostino1998 *Studies on the Effectiveness of Antihistamines in Reducing Symptoms of the Common Cold*

Description

Results from 9 studies on the effectiveness of antihistamines in reducing the severity of runny nose and sneezing in the common cold.

Usage

```
dat.dagostino1998
```

Format

The data frame contains the following columns:

study	`numeric`	study id
cold	`character`	natural or induced cold study
scale.rn	`character`	scale for measuring runny nose severity
scale.sn	`character`	scale for measuring sneezing severity
drug	`character`	type of antihistamine studied
tnt	`numeric`	total sample size of the treatment group
tnc	`numeric`	total sample size of the control (placebo) group
outcome	`character`	outcome variable (see 'Details')
mt	`numeric`	mean in the treatment group
sdt	`numeric`	SD in the treatment group
mc	`numeric`	mean in the control group
sdc	`numeric`	SD in the control group
xt	`numeric`	number of patients reaching the therapy goal in the treatment group
xc	`numeric`	number of patients reaching the therapy goal in the control (placebo) group
nt	`numeric`	sample size of the treatment group for measuring the outcome
nc	`numeric`	sample size of the control group for measuring the outcome

Details

The studies for this meta-analysis were assembled to examine the effectiveness of antihistamines in reducing the severity of runny nose and sneezing in the common cold. Effectiveness was measured after one and two days of treatment in terms of 4 different outcome variables:

1. `rnic1` and `rnic2` (continuous): incremental change (improvement) in runny nose severity at day 1 and day 2,
2. `rngoal1` and `rngoal2` (dichotomous): reaching the goal of therapy (of at least a 50% reduction in runny nose severity) at day 1 and day 2,
3. `snic1` and `snic2` (continuous): incremental change (improvement) in sneezing severity at day 1 and day 2, and
4. `rngoal1` and `rngoal2` (dichotomous): reaching the goal of therapy (of at least a 50% reduction in sneezing severity) at day 1 and day 2.

For the continuous outcomes, standardized mean differences can be computed to quantify the difference between the treatment and control groups. For the dichotomous outcomes, one can compute (log) odds ratios to quantify the difference between the treatment and control groups.

Source

D'Agostino, R. B., Sr., Weintraub, M., Russell, H. K., Stepanians, M., D'Agostino, R. B., Jr., Cantilena, L. R., Jr., Graumlich, J. F., Maldonado, S., Honig, P., & Anello, C. (1998). The effectiveness of antihistamines in reducing the severity of runny nose and sneezing: A meta-analysis. *Clinical Pharmacology & Therapeutics*, **64**(6), 579–596. https://doi.org/10.1016/S0009-9236(98)90049-2

Examples

```
### copy data into 'dat' and examine data
dat <- dat.dagostino1998
dat

### compute standardized mean differences and corresponding sampling variances
dat <- escalc(measure="SMD", m1i=mt, m2i=mc, sd1i=sdt, sd2i=sdc, n1i=nt, n2i=nc, data=dat,
              add.measure=TRUE)

### compute log odds ratios and corresponding sampling variances
dat <- escalc(measure="OR",  ai=xt, ci=xc, n1i=nt, n2i=nc, data=dat,
              replace=FALSE, add.measure=TRUE, add=1/2, to="all")

### inspect data for the first study
head(dat, 8)

### fit a random-effects model for incremental change in runny nose severity at day 1
res <- rma(yi, vi, data=dat, subset=outcome=="rnic1")
res

### fit a random-effects model for reaching the goal of therapy for runny nose severity at day 1
res <- rma(yi, vi, data=dat, subset=outcome=="rngoal1")
res
predict(res, transf=exp)

### construct approximate V matrix assuming a correlation of 0.7 for sampling errors within studies
V <- lapply(split(dat$vi, dat$study), function(v) {
   S <- diag(sqrt(v), nrow=length(v), ncol=length(v))
   R <- matrix(0.7, nrow=length(v), ncol=length(v))
   diag(R) <- 1
   S %*% R %*% S
})
V <- bldiag(V, order=dat$study)

### fit a model for incremental change in runny nose severity at day 1 and at day 2, allowing for
### correlated sampling errors (no random effects added, since there does not appear to be any
### noteworthy heterogeneity in these data)
res <- rma.mv(yi, V, mods = ~ outcome - 1, data=dat, subset=outcome %in% c("rnic1","rnic2"))
res

### test if there is a difference in effects at day 1 and day 2
anova(res, X=c(1,-1))
```

dat.damico2009 *Studies on Topical plus Systemic Antibiotics to Prevent Respiratory Tract Infections*

Description

Results from 16 studies examining the effectiveness of topical plus systemic antibiotics to prevent respiratory tract infections (RTIs).

Usage

```
dat.damico2009
```

Format

The data frame contains the following columns:

study	`character`	first author
year	`numeric`	publication year
xt	`numeric`	number of RTIs in the treatment group
nt	`numeric`	number of patients in the treatment group
xc	`numeric`	number of RTIs in the control group
nc	`numeric`	number of patients in the control group
conceal	`numeric`	allocation concealment (0 = not adequate, 1 = adequate)
blind	`numeric`	blinding (0 = open, 1 = double-blind)

Details

The dataset includes the results from 16 studies that examined the effectiveness of topical plus systemic antibiotics versus no prophylaxis to prevent respiratory tract infections (RTIs).

Source

D'Amico, R., Pifferi, S., Torri, V., Brazzi, L., Parmelli, E., & Liberati, A. (2009). Antibiotic prophylaxis to reduce respiratory tract infections and mortality in adults receiving intensive care. *Cochrane Database of Systematic Reviews*, **4**, CD000022. https://doi.org/10.1002/14651858.CD000022.pub3

Examples

```
### copy data into 'dat' and examine data
dat <- dat.damico2009
dat

### meta-analysis of the (log) odds ratios using the Mantel-Haenszel method
rma.mh(measure="OR", ai=xt, n1i=nt, ci=xc, n2i=nc, data=dat, digits=2)

### calculate log odds ratios and corresponding sampling variances
dat <- escalc(measure="OR", ai=xt, n1i=nt, ci=xc, n2i=nc, data=dat)
```

```
### meta-analysis using a random-effects model
res <- rma(yi, vi, data=dat, method="DL")
res
predict(res, transf=exp, digits=2)
```

---

dat.debruin2009 *Studies on Standard Care Quality and HAART-Adherence*

Description

Results from 13 trials providing information about standard care quality and HAART-adherence in control groups.

Usage

```
dat.debruin2009
```

Format

The data frame contains the following columns:

author	`character`	(first) author of study
year	`numeric`	publication year
scq	`numeric`	standard care quality
ni	`numeric`	number of patients in the standard care group
xi	`numeric`	number of patients with an undetectable viral load in standard care group
mi	`numeric`	number of patients with a detectable viral load in standard care group
ethnicity	`character`	dominant ethnicity of the patients in the standard care group
patients	`character`	inclusion of patients continuing or starting (a new) treatment
select	`character`	baseline selection of patients with adherence problems or no selection
sens	`character`	sensitivity of viral load assessments (<400 vs. >=400 copies/ml)

Details

Highly active antiretroviral therapy (HAART) refers to a combination of multiple antiretroviral drugs that can effectively suppress the HIV virus. However, achieving viral suppression (to the point that the virus becomes essentially undetectable in a blood sample) requires high levels of adherence to an often complicated medication regimen. A number of trials have examined various interventions that aim to increase adherence levels. In each trial, patients receiving the intervention are compared to patients in a control group receiving standard care (often referred to as ‘care as usual’). However, the quality of standard care can vary substantially between these studies. de Bruin et al. (2009) assessed the quality of standard care provided (based on a quantification of the number of behavior change techniques applied) and examined to what extent the quality of standard care was related to the proportion of patients achieving effective viral suppression in the control groups.

Source

de Bruin, M., Viechtbauer, W., Hospers, H. J., Schaalma, H. P., & Kok, G. (2009). Standard care quality determines treatment outcomes in control groups of HAART-adherence intervention studies: Implications for the interpretation and comparison of intervention effects. *Health Psychology*, **28**(6), 668–674. https://doi.org/10.1037/a0015989

Examples

```
### copy data into 'dat'
dat <- dat.debruin2009

### calculate proportions and corresponding sampling variances
dat <- escalc(measure="PR", xi=xi, ni=ni, data=dat)
dat

### random-effects model
res <- rma(yi, vi, data=dat)
print(res, digits=2)

### mixed-effects meta-regression model with all predictors/covariates
res <- rma(yi, vi, mods = ~ scq + ethnicity + patients + select + sens, data=dat)
print(res, digits=3)

### mixed-effects meta-regression model with scq and ethnicity as predictors/covariates
res <- rma(yi, vi, mods = ~ scq + ethnicity, data=dat)
print(res, digits=3)
```

dat.dorn2007 *Studies on Complementary and Alternative Medicine for Irritable Bowel Syndrome*

Description

Results from 19 trials examining complementary and alternative medicine (CAM) for irritable bowel syndrome (IBS).

Usage

```
dat.dorn2007
```

Format

The data frame contains the following columns:

id	`numeric`	trial id number
study	`character`	(first) author
year	`numeric`	publication year
country	`character`	country where trial was conducted

ibs.crit	`character`	IBS diagnostic criteria (Manning, Rome I, Rome II, or Other)
days	`numeric`	number of treatment days
visits	`numeric`	number of practitioner visits
jada	`numeric`	Jadad score
x.a	`numeric`	number of responders in the active treatment group
n.a	`numeric`	number of participants in the active treatment group
x.p	`numeric`	number of responders in the placebo group
n.p	`numeric`	number of participants in the placebo group

Details

The dataset includes the results from 19 randomized clinical trials that examined the effectiveness of complementary and alternative medicine (CAM) for irritable bowel syndrome (IBS).

Note

The data were extracted from Table I in Dorn et al. (2009). Comparing the funnel plot in Figure 1 with the one obtained below indicates that the data for study 5 (Davis et al., 2006) in the table were not the ones that were used in the actual analyses.

Source

Dorn, S. D., Kaptchuk, T. J., Park, J. B., Nguyen, L. T., Canenguez, K., Nam, B. H., Woods, K. B., Conboy, L. A., Stason, W. B., & Lembo, A. J. (2007). A meta-analysis of the placebo response in complementary and alternative medicine trials of irritable bowel syndrome. *Neurogastroenterology & Motility*, **19**(8), 630–637. https://doi.org/10.1111/j.1365-2982.2007.00937.x

Examples

```
### copy data into 'dat' and examine data
dat <- dat.dorn2007
dat

### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=x.a, n1i=n.a, ci=x.p, n2i=n.p, data=dat)

### random-effects model
res <- rma(yi, vi, data=dat, digits=2, method="DL")
res

### estimated average risk ratio
predict(res, transf=exp)

### funnel plot with study 5 highlighted in red
funnel(res, atransf=exp, at=log(c(.1, .2, .5, 1, 2, 5, 10)),
       ylim=c(0,1), steps=6, las=1, col=ifelse(dat$id == 5, "red", "black"))

### change log risk ratio for study 5
dat$yi[5] <- -0.44

### results are now more in line with what is reported in the paper
### (although the CI in the paper is not wide enough)
res <- rma(yi, vi, data=dat, digits=2, method="DL")
predict(res, transf=exp)

### funnel plot with study 5 highlighted in red
funnel(res, atransf=exp, at=log(c(.1, .2, .5, 1, 2, 5, 10)),
       ylim=c(0,1), steps=6, las=1, col=ifelse(dat$id == 5, "red", "black"))
```

`dat.egger2001` *Studies on the Effectiveness of Intravenous Magnesium in Acute Myocardial Infarction*

Description

Results from 16 trials examining the effectiveness of intravenous magnesium in the prevention of death following acute myocardial infarction.

Usage

```
dat.egger2001
```

Format

The data frame contains the following columns:

id	`numeric`	trial id number
study	`character`	first author or trial name
year	`numeric`	publication year
ai	`numeric`	number of deaths in the magnesium group
n1i	`numeric`	number of patients in the magnesium group
ci	`numeric`	number of deaths in the control group
n2i	`numeric`	number of patients in the control group

Details

The dataset includes the results from 16 randomized clinical trials that examined the effectiveness of intravenous magnesium in the prevention of death following acute myocardial infarction. Studies 1-7 were included in the meta-analyses by Teo et al. (1991) and Horner (1992) and were combined with the results from the LIMIT-2 trial (Woods et al., 1992) in Yusuf et al. (1993), suggesting that magnesium is an effective treatment for reducing mortality. However, the results from the ISIS-4 mega trial (ISIS-4 Collaborative Group, 1995) indicated no reduction in mortality with magnesium treatment. Publication bias has been suggested as one possible explanation for the conflicting findings (Egger & Davey Smith, 1995).

The present dataset includes some additional trials and are based on Table 18.2 from Egger, Davey Smith, and Altman (2001).

Source

Egger, M., Davey Smith, G., & Altman, D. G. (Eds.) (2001). *Systematic reviews in health care: Meta-analysis in context* (2nd ed.). London: BMJ Books.

References

Egger, M., & Davey Smith, G. (1995). Misleading meta-analysis: Lessons from "an effective, safe, simple" intervention that wasn't. *British Medical Journal*, **310**(6982), 752–754. https://doi.org/10.1136/bmj.310.6982.752

Horner, S. M. (1992). Efficacy of intravenous magnesium in acute myocardial infarction in reducing arrhythmias and mortality: Meta-analysis of magnesium in acute myocardial infarction. *Circulation*, **86**(3), 774–779. https://doi.org/10.1161/01.cir.86.3.774

ISIS-4 Collaborative Group (1995). ISIS-4: A randomised factorial trial assessing early oral captopril, oral mononitrate, and intravenous magnesium sulphate in 58,050 patients with suspected acute myocardial infarction. *Lancet*, **345**(8951), 669–685. https://doi.org/10.1016/S0140-6736(95)90865-X

Teo, K. K., Yusuf, S., Collins, R., Held, P. H., & Peto, R. (1991). Effects of intravenous magnesium in suspected acute myocardial infarction: Overview of randomised trials. *British Medical Journal*, **303**(6816), 1499–1503. https://doi.org/10.1136/bmj.303.6816.1499

Woods, K. L., Fletcher, S., Roffe, C., & Haider, Y. (1992). Intravenous magnesium sulphate in suspected acute myocardial infarction: Results of the second Leicester Intravenous Magnesium Intervention Trial (LIMIT-2). *Lancet*, **339**(8809), 1553–1558. https://doi.org/10.1016/0140-6736(92)91828-v

Yusuf, S., Teo, K., & Woods, K. (1993). Intravenous magnesium in acute myocardial infarction: An effective, safe, simple, and inexpensive treatment. *Circulation*, **87**(6), 2043–2046. https://doi.org/10.1161/01.cir.87.6.2043

See Also

dat.li2007

Examples

```
### copy data into 'dat' and examine data
dat <- dat.egger2001
dat

### meta-analysis of trials 1-7 using Peto's method (as in Teo et al., 1991)
res <- rma.peto(ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, subset=1:7)
print(res, digits=2)

### meta-analysis of trials 1-7 and LIMIT-2 (as in Yusuf et al., 1993)
res <- rma.peto(ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, subset=c(1:7,14))
print(res, digits=2)

### meta-analysis of all trials except ISIS-4
res <- rma.peto(ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, subset=-16)
print(res, digits=2)
predict(res, transf=exp, digits=2)
```

```
### meta-analysis of all trials including ISIS-4
res <- rma.peto(ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat)
print(res, digits=2)
predict(res, transf=exp, digits=2)

### contour-enhanced funnel plot centered at 0
funnel(res, refline=0, level=c(90, 95, 99), shade=c("white", "gray", "darkgray"))
```

`dat.fine1993` *Studies on Radiation Therapy with or without Adjuvant Chemotherapy in Patients with Malignant Gliomas*

Description

Results from 17 trials comparing post-operative radiation therapy with and without adjuvant chemotherapy in patients with malignant gliomas.

Usage

```
dat.fine1993
```

Format

The data frame contains the following columns:

study	`numeric`	study number
nei	`numeric`	sample size in the experimental group receiving radiotherapy plus adjuvant chemotherapy
nci	`numeric`	sample size in the control group receiving radiotherapy alone
e1i	`numeric`	number of survivors at 6 months in the experimental group
c1i	`numeric`	number of survivors at 6 months in the control group
e2i	`numeric`	number of survivors at 12 months in the experimental group
c2i	`numeric`	number of survivors at 12 months in the control group
e3i	`numeric`	number of survivors at 18 months in the experimental group
c3i	`numeric`	number of survivors at 18 months in the control group
e4i	`numeric`	number of survivors at 24 months in the experimental group
c4i	`numeric`	number of survivors at 24 months in the control group

Details

The 17 trials report the post-operative survival of patients with malignant gliomas receiving either radiation therapy with adjuvant chemotherapy or radiation therapy alone. Survival was assessed at 6, 12, 18, and 24 months in all but one study (which assessed survival only at 12 and at 24 months).

The data were reconstructed by Trikalinos and Olkin (2012) based on Table 2 in Fine et al. (1993) and Table 3 in Dear (1994). The data can be used to illustrate how a meta-analysis can be conducted of effect sizes reported at multiple time points using a multivariate model.

Source

Dear, K. B. G. (1994). Iterative generalized least squares for meta-analysis of survival data at multiple times. *Biometrics*, **50**(4), 989–1002. https://doi.org/10.2307/2533438

Trikalinos, T. A., & Olkin, I. (2012). Meta-analysis of effect sizes reported at multiple time points: A multivariate approach. *Clinical Trials*, **9**(5), 610–620. https://doi.org/10.1177/1740774512453218

References

Fine, H. A., Dear, K. B., Loeffler, J. S., Black, P. M., & Canellos, G. P. (1993). Meta-analysis of radiation therapy with and without adjuvant chemotherapy for malignant gliomas in adults. *Cancer*, **71**(8), 2585–2597. https://doi.org/10.1002/1097-0142(19930415)71:8<2585::aid-cncr2820710825>3.0.co;2-s

Examples

```
### copy data into 'dat' and examine data
dat <- dat.fine1993
dat

### calculate log(ORs) and sampling variances for each time point
dat <- escalc(measure=”OR”, ai=e1i, n1i=nei, ci=c1i, n2i=nci, data=dat, var.names=c(”y1i”,”v1i”))
dat <- escalc(measure=”OR”, ai=e2i, n1i=nei, ci=c2i, n2i=nci, data=dat, var.names=c(”y2i”,”v2i”))
dat <- escalc(measure=”OR”, ai=e3i, n1i=nei, ci=c3i, n2i=nci, data=dat, var.names=c(”y3i”,”v3i”))
dat <- escalc(measure=”OR”, ai=e4i, n1i=nei, ci=c4i, n2i=nci, data=dat, var.names=c(”y4i”,”v4i”))

### calculate the covariances (equations in Appendix of Trikalinos & Olkin, 2012)
dat$v12i <- with(dat, nei / (e1i * (nei - e2i)) + nci / (c1i * (nci - c2i)))
dat$v13i <- with(dat, nei / (e1i * (nei - e3i)) + nci / (c1i * (nci - c3i)))
dat$v14i <- with(dat, nei / (e1i * (nei - e4i)) + nci / (c1i * (nci - c4i)))
dat$v23i <- with(dat, nei / (e2i * (nei - e3i)) + nci / (c2i * (nci - c3i)))
dat$v24i <- with(dat, nei / (e2i * (nei - e4i)) + nci / (c2i * (nci - c4i)))
dat$v34i <- with(dat, nei / (e3i * (nei - e4i)) + nci / (c3i * (nci - c4i)))

### create dataset in long format
dat.long <- data.frame(study=rep(1:nrow(dat), each=4), time=1:4,
                       yi=c(t(dat[c(”y1i”,”y2i”,”y3i”,”y4i”)])),
                       vi=c(t(dat[c(”v1i”,”v2i”,”v3i”,”v4i”)])))

### var-cov matrices of the sudies
V <- lapply(split(dat, dat$study),
            function(x) matrix(c( x$v1i, x$v12i, x$v13i, x$v14i,
                                 x$v12i,  x$v2i, x$v23i, x$v24i,
                                 x$v13i, x$v23i,  x$v3i, x$v34i,
                            x$v14i, x$v24i, x$v34i,  x$v4i), nrow=4, ncol=4, byrow=TRUE))

### remove rows for the missing time points in study 17
dat.long <- na.omit(dat.long)

### remove corresponding rows/columns from var-cov matrix
V[[17]] <- V[[17]][c(2,4),c(2,4)]
```

```
### make a copy of V
Vc <- V

### replace any (near) singular var-cov matrices with ridge corrected versions
repl.Vi <- function(Vi) {
   res <- eigen(Vi)
   if (any(res$values <= .08)) {
      round(res$vectors %*% diag(res$values + .08) %*% t(res$vectors), 12)
   } else {
      Vi
   }
}
Vc <- lapply(Vc, repl.Vi)

### do not correct var-cov matrix of study 17
Vc[[17]] <- V[[17]]

### construct block diagonal matrix
Vc <- bldiag(Vc)

### multivariate fixed-effects model
res <- rma.mv(yi, Vc, mods = ~ factor(time) - 1, method="FE", data=dat.long)
print(res, digits=3)

### multivariate random-effects model with heteroscedastic AR(1) structure for the true effects
res <- rma.mv(yi, Vc, mods = ~ factor(time) - 1, random = ~ time | study,
              struct="HAR", data=dat.long)
print(res, digits=3)

## Not run:
### profile the variance components
par(mfrow=c(2,2))
profile(res, tau2=1, xlim=c( 0,.2))
profile(res, tau2=2, xlim=c( 0,.2))
profile(res, tau2=3, xlim=c( 0,.2))
profile(res, tau2=4, xlim=c(.1,.3))
## End(Not run)

## Not run:
### profile the autocorrelation coefficient
par(mfrow=c(1,1))
profile(res, rho=1)
## End(Not run)
```

dat.gibson2002 *Studies on the Effectiveness of Self-Management Education and Regular Medical Review for Adults with Asthma*

Description

Results from 15 trials examining the effectiveness of self-management education and regular medical review for adults with asthma.

Usage

```
dat.gibson2002
```

Format

The data frame contains the following columns:

author	`character`	first author of study
year	`numeric`	publication year
n1i	`numeric`	number of participants in the intervention group
m1i	`numeric`	mean number of days off work/school in the intervention group
sd1i	`numeric`	standard deviation of the number of days off work/school in the intervention group
n2i	`numeric`	number of participants in the control/comparison group
m2i	`numeric`	mean number of days off work/school in the control/comparison group
sd2i	`numeric`	standard deviation of the number of days off work/school in the control/comparison group
ai	`numeric`	number of participants who had one or more days off work/school in the intervention group
bi	`numeric`	number of participants who no days off work/school in the intervention group
ci	`numeric`	number of participants who had one or more days off work/school in the control/comparison group
di	`numeric`	number of participants who no days off work/school in the control/comparison group
type	`numeric`	numeric code for the intervention type (see 'Details')

Details

Asthma management guidelines typically recommend for patients to receive education and regular medical review. While self-management programs have been shown to increase patient knowledge, it is less clear to what extent they actually impact health outcomes. The systematic review by Gibson et al. (2002) examined the effectiveness of self-management education and regular medical review for adults with asthma. In each study, participants receiving a certain management intervention were compared against those in a control/comparison group with respect to a variety of health outcomes. One of the outcomes examined in a number of studies was the number of days off work/school.

The majority of studies reporting this outcome provided means and standard deviations allowing a meta-analysis of standardized mean differences. Seven studies also reported the number of participants who had one or more days off work/school in each group. These studies could be meta-analyzed using, for example, (log) risk ratios. Finally, one could also consider a combined analysis based on standardized mean differences computed from the means and standard deviations where available and using probit transformed risk differences (which also provide estimates of the standardized mean difference) for the remaining studies.

Some degree of patient education was provided in all studies. In addition, the `type` variable indicates what additional intervention components were included in each study:

1. optimal self-management (writing action plan, self-monitoring, regular medical review),
2. self-monitoring and regular medical review,
3. self-monitoring only,

4. regular medical review only,
5. written action plan only.

Source

Gibson, P. G., Powell, H., Wilson, A., Abramson, M. J., Haywood, P., Bauman, A., Hensley, M. J., Walters, E. H., & Roberts, J. J. L. (2002). Self-management education and regular practitioner review for adults with asthma. *Cochrane Database of Systematic Reviews*, **3**, CD001117. https://doi.org/10.1002/14651858.CD001117

Examples

```
### copy data into 'dat' and examine data
dat <- dat.gibson2002
dat

### fixed-effects model analysis of the standardized mean differences
dat <- escalc(measure="SMD", m1i=m1i, sd1i=sd1i, n1i=n1i, m2i=m2i, sd2i=sd2i, n2i=n2i, data=dat)
res <- rma(yi, vi, data=dat, method="FE")
print(res, digits=2)

### fixed-effects model analysis of the (log) risk ratios
dat <- escalc(measure="RR", ai=ai, bi=bi, ci=ci, di=di, data=dat)
res <- rma(yi, vi, data=dat, method="FE")
print(res, digits=2)
predict(res, transf=exp, digits=2)

### fixed-effects model analysis of the standardized mean differences and the probit transformed
### risk differences (which also provide estimates of the standardized mean difference)
dat <- escalc(measure="SMD", m1i=m1i, sd1i=sd1i, n1i=n1i, m2i=m2i, sd2i=sd2i, n2i=n2i, data=dat)
dat <- escalc(measure="PBIT", ai=ai, bi=bi, ci=ci, di=di, data=dat, replace=FALSE)
dat
res <- rma(yi, vi, data=dat, method="FE")
print(res, digits=2)
```

dat.graves2010 *Studies on the Effectiveness of Injected Cholera Vaccines*

Description

Results from 17 studies on the effectiveness of injected vaccines against cholera.

Usage

```
dat.graves2010
```

Format

The data frame contains the following columns:

study	`character`	author/study name and publication year
ai	`numeric`	number of cholera cases in the vaccinated group
n1i	`numeric`	number of individuals in the vaccinated group
ci	`numeric`	number of cholera cases in the placebo group
n2i	`numeric`	number of individuals in the placebo group

Details

Cholera is an infection caused by certain strains of the bacterium *Vibrio cholerae*. When untreated, mortality rates can be as high as 50-60%. Proper sanitation practices are usually effective in preventing outbreaks, but a number of oral and injectable vaccines have also been developed. The Cochrane review by Graves et al. (2010) examined the effectiveness of injectable vaccines for preventing cholera cases and death. The present dataset includes results from 17 studies that reported the number of cholera cases in vaccinated and placebo/comparison groups up to 7 months after the treatment.

Source

Graves, P. M., Deeks, J. J., Demicheli, V., & Jefferson, T. (2010). Vaccines for preventing cholera: Killed whole cell or other subunit vaccines (injected). *Cochrane Database of Systematic Reviews*, **8**, CD000974. https://doi.org/10.1002/14651858.CD000974.pub2

Examples

```
### copy data into 'dat' and examine data
dat <- dat.graves2010
dat

### analysis using the Mantel-Haenszel method
rma.mh(measure="RR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, digits=2)
```

`dat.hackshaw1998` *Studies on the Risk of Lung Cancer in Women Exposed to Environmental Tobacco Smoke*

Description

Results from 37 studies on the risk of lung cancer in women exposed to environmental tobacco smoke (ETS) from their smoking spouse.

Usage

```
dat.hackshaw1998
```

Format

The data frame contains the following columns:

study	`numeric`	study number
author	`character`	first author of study
year	`numeric`	publication year
country	`character`	country where study was conducted
design	`character`	study design (either cohort or case-control)
cases	`numeric`	number of lung cancer cases
or	`numeric`	odds ratio
or.lb	`numeric`	lower bound of 95% CI for the odds ratio
or.ub	`numeric`	upper bound of 95% CI for the odds ratio
yi	`numeric`	log odds ratio
vi	`numeric`	corresponding sampling variance

Details

The dataset includes the results from 37 studies (4 cohort, 33 case-control) examining if women (who are lifelong nonsmokers) have an elevated risk for lung cancer due to exposure to environmental tobacco smoke (ETS) from their smoking spouse. Values of the log odds ratio greater than 0 indicate an increased risk of cancer in exposed women compared to women not exposed to ETS from their spouse.

Note that the log odds ratios and corresponding sampling variances were back-calculated from the reported odds ratios and confidence interval (CI) bounds (see ‘Examples’). Since the reported values were rounded to some extent, this introduces some minor inaccuracies into the back-calculations. The overall estimate reported in Hackshaw et al. (1997) and Hackshaw (1998) can be fully reproduced though.

Source

Hackshaw, A. K., Law, M. R., & Wald, N. J. (1997). The accumulated evidence on lung cancer and environmental tobacco smoke. *British Medical Journal*, **315**(7114), 980–988. https://doi.org/10.1136/bmj.315.7114.980

Hackshaw, A. K. (1998). Lung cancer and passive smoking. *Statistical Methods in Medical Research*, **7**(2), 119–136. https://doi.org/10.1177/096228029800700203

Examples

```
### copy data into 'dat' and examine data
dat <- dat.hackshaw1998
dat

### random-effects model using the log odds ratios
res <- rma(yi, vi, data=dat, method="DL")
res

### estimated average odds ratio with CI (and prediction interval)
predict(res, transf=exp, digits=2)

### illustrate how the log odds ratios and corresponding sampling variances
### were back-calculated based on the reported odds ratios and CI bounds
dat$yi <- log(dat$or)
dat$vi <- ((log(dat$or.ub) - log(dat$or.lb)) / (2*qnorm(.975)))^2
```

`dat.hahn2001` *Studies on the Effectiveness of Different Rehydration Solutions for the Prevention of Unscheduled Intravenous Infusion in Children with Diarrhoea*

Description

Results from 12 trials examining the effectiveness of a reduced versus standard rehydration solution for the prevention of unscheduled intravenous infusion in children with diarrhoea.

Usage

```
dat.hahn2001
```

Format

The data frame contains the following columns:

study	`character`	trial name and year
ai	`numeric`	number of children requiring unscheduled intravenous infusion in the reduced rehydration solution grou
n1i	`numeric`	number of children in the reduced rehydration solution group
ci	`numeric`	number of children requiring unscheduled intravenous infusion in the standard rehydration solution grou
n2i	`numeric`	number of children in the standard rehydration solution group

Details

The dataset includes the results from 12 randomized clinical trials that examined the effectiveness of a reduced osmolarity oral rehydration solution (total osmolarity <250 mmol/l with reduced sodium) with a standard WHO oral rehydration solution (sodium 90 mmol/l, glucose 111mmol/l, total osmolarity 311 mmol/l) for the prevention of unscheduled intravenous infusion in children with diarrhoea.

Source

Hahn, S., Kim, Y., & Garner, P. (2001). Reduced osmolarity oral rehydration solution for treating dehydration due to diarrhoea in children: Systematic review. *British Medical Journal*, **323**(7304), 81–85. https://doi.org/10.1136/bmj.323.7304.81

Examples

```
### copy data into 'dat' and examine data
dat <- dat.hahn2001
dat

### meta-analysis of (log) odds rations using the Mantel-Haenszel method
res <- rma.mh(measure=”OR”, ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, digits=2, slab=study)
res
```

```
### forest plot (also show studies that were excluded from the analysis)
options(na.action="na.pass")
forest(res, atransf=exp, at=log(c(.01, .1, 1, 10, 100)), header=TRUE)
options(na.action="na.omit")
```

dat.hart1999 *Studies on the Effectiveness of Warfarin for Preventing Strokes*

Description

Results from 6 clinical trials examining the effectiveness of adjusted-dose warfarin for preventing strokes in patients with atrial fibrillation.

Usage

```
dat.hart1999
```

Format

The data frame contains the following columns:

trial	`numeric`	trial number
study	`character`	study name (abbreviated)
year	`numeric`	publication year
x1i	`numeric`	number of strokes in the warfarin group
n1i	`numeric`	number of patients in the warfarin group
t1i	`numeric`	total person-time (in years) in the warfarin group
x2i	`numeric`	number of strokes in the placebo/control group
n2i	`numeric`	number of patients in the placebo/control group
t2i	`numeric`	total person-time (in years) in the placebo/control group
compgrp	`character`	type of comparison group (placebo or control)
prevtype	`character`	type of prevention (primary or secondary)
trinr	`character`	target range for the international normalized ratio (INR)

Details

The 6 studies provide data with respect to the number of strokes in the warfarin and the comparison (placebo or control) group. In addition, the number of patients and the total person-time (in years) is provided for the two groups. The goal of the meta-analysis was to examine the effectiveness of adjusted-dose warfarin for preventing strokes in patients with atrial fibrillation.

Source

Hart, R. G., Benavente, O., McBride, R., & Pearce, L. A. (1999). Antithrombotic therapy to prevent stroke in patients with atrial fibrillation: A meta-analysis. *Annals of Internal Medicine*, **131**(7), 492–501. https://doi.org/10.7326/0003-4819-131-7-199910050-00003

Examples

```
### copy data into 'dat'
dat <- dat.hart1999

### calculate log incidence rate ratios and corresponding sampling variances
dat <- escalc(measure="IRR", x1i=x1i, x2i=x2i, t1i=t1i, t2i=t2i, data=dat)
dat

### meta-analysis of log incidence rate ratios using a random-effects model
res <- rma(yi, vi, data=dat)
res

### average incidence rate ratio with 95% CI
predict(res, transf=exp)

### forest plot with extra annotations
par(mar=c(5,4,1,2))
forest(res, xlim=c(-11, 5), at=log(c(.05, .25, 1, 4)), atransf=exp,
       slab=paste0(dat$study, " (", dat$year, ")"),
       ilab=cbind(paste(dat$x1i, "/", dat$t1i, sep=" "),
       paste(dat$x2i, "/", dat$t2i, sep=" ")),
       ilab.xpos=c(-6.5,-4), cex=.85, header="Study (Year)")
op <- par(cex=.85, font=2)
text(c(-6.5,-4), 8.5, c("Warfarin", "Control"))
text(c(-6.5,-4), 7.5, c("Strokes / PT", "Strokes / PT"))
segments(x0=-8, y0=8, x1=-2.75, y1=8)
par(op)

### meta-analysis of incidence rate differences using a random-effects model
res <- rma(measure="IRD", x1i=x1i, x2i=x2i, t1i=t1i, t2i=t2i, data=dat)
res
```

`dat.hasselblad1998` *Studies on the Effectiveness of Counseling for Smoking Cessation*

Description

Results from 24 studies on the effectiveness of various counseling types for smoking cessation.

Usage

```
dat.hasselblad1998
```

Format

The data frame contains the following columns:

id	`numeric`	id number for each treatment arm
study	`numeric`	study id number

authors	`character`	study author(s)
year	`numeric`	publication year
trt	`character`	intervention group
xi	`numeric`	number of individuals abstinent
ni	`numeric`	number of individuals in group

Details

The dataset includes the results from 24 studies on the effectiveness of various counseling types for smoking cessation (i.e., self-help, individual counseling, group counseling, and no contact). The dataset indicates the total number of individuals within each study arm and the number that were abstinent from 6 to 12 months. The majority of the studies compared two interventions types against each other, while 2 studies compared three types against each other simultaneously.

The data can be used for a 'network meta-analysis' (also called a 'mixed treatment comparison'). The code below shows how such an analysis can be conducted using an arm-based and a contrast-based model (see Salanti et al., 2008, for more details).

Source

Hasselblad, V. (1998). Meta-analysis of multitreatment studies. *Medical Decision Making*, **18**(1), 37–43. https://doi.org/10.1177/0272989X9801800110

References

Gleser, L. J., & Olkin, I. (2009). Stochastically dependent effect sizes. In H. Cooper, L. V. Hedges, & J. C. Valentine (Eds.), *The handbook of research synthesis and meta-analysis* (2nd ed., pp. 357–376). New York: Russell Sage Foundation.

Law, M., Jackson, D., Turner, R., Rhodes, K., & Viechtbauer, W. (2016). Two new methods to fit models for network meta-analysis with random inconsistency effects. *BMC Medical Research Methodology*, **16**, 87. https://doi.org/10.1186/s12874-016-0184-5

Salanti, G., Higgins, J. P. T., Ades, A. E., & Ioannidis, J. P. A. (2008). Evaluation of networks of randomized trials. *Statistical Methods in Medical Research*, **17**(3), 279–301. https://doi.org/10.1177/0962280207080643

Examples

```
### copy data into 'dat'
dat <- dat.hasselblad1998
dat

### create network graph ('igraph' package must be installed)
## Not run:
require(igraph)
pairs <- data.frame(do.call(rbind,
  sapply(split(dat$trt, dat$study), function(x) t(combn(x,2)))), stringsAsFactors=FALSE)
lvls <- c("no_contact", "self_help", "ind_counseling", "grp_counseling")
pairs$X1 <- factor(pairs$X1, levels=lvls)
pairs$X2 <- factor(pairs$X2, levels=lvls)
tab <- table(pairs[,1], pairs[,2])
tab # adjacency matrix
```

```
g <- graph_from_adjacency_matrix(tab, mode = "plus", weighted=TRUE, diag=FALSE)
vertex_attr(g, "name") <- c("No Contact", "Self-Help",
                            "Individual\nCounseling", "Group\nCounseling")
plot(g, edge.curved=FALSE, edge.width=E(g)$weight, layout=layout_on_grid,
   vertex.size=45, vertex.color="lightgray", vertex.label.color="black", vertex.label.font=2)
## End(Not run)

### calculate log odds for each study arm
dat <- escalc(measure="PLO", xi=xi, ni=ni, add=1/2, to="all", data=dat)
dat

### convert trt variable to factor with desired ordering of levels
dat$trt <- factor(dat$trt, levels=c("no_contact", "self_help", "ind_counseling", "grp_counseling"))

### add a space before each level (this makes the output a bit more legible)
levels(dat$trt) <- paste0(" ", levels(dat$trt))

### network meta-analysis using an arm-based model with fixed study effects
### by setting rho=1/2, tau^2 reflects the amount of heterogeneity for all treatment comparisons
res <- rma.mv(yi, vi, mods = ~ factor(study) + trt - 1,
              random = ~ trt | study, rho=1/2, data=dat, btt="trt")
res

### all pairwise odds ratios of interventions versus no contact
predict(res, newmods=cbind(matrix(0, nrow=3, ncol=24), diag(3)),
        intercept=FALSE, transf=exp, digits=2)

### all pairwise odds ratios comparing interventions (ic vs sh, gc vs sh, and gc vs ic)
predict(res, newmods=cbind(matrix(0, nrow=3, ncol=24), rbind(c(-1,1,0), c(-1,0,1), c(0,-1,1))),
        intercept=FALSE, transf=exp, digits=2)

### forest plot of ORs of interventions versus no contact
dev.new(width=7, height=4)
par(mar=c(5,4,1,2))
forest(c(0,res$beta[25:27]), sei=c(0,res$se[25:27]), psize=1, xlim=c(-3,4), digits=c(2,1), efac=2,
       slab=c("No Contact", "Self-Help", "Individual Counseling", "Group Counseling"),
     atransf=exp, at=log(c(.5, 1, 2, 4, 8)), xlab="Odds Ratio for Intervention vs. No Contact",
       header=c("Intervention", "Odds Ratio [95% CI]"))

############################################################################

### restructure dataset to a contrast-based format
dat <- to.wide(dat.hasselblad1998, study="study", grp="trt", ref="no_contact", grpvars=6:7)

### calculate log odds ratios for each treatment comparison
dat <- escalc(measure="OR", ai=xi.1, n1i=ni.1,
                            ci=xi.2, n2i=ni.2, add=1/2, to="all", data=dat)
dat

### calculate the variance-covariance matrix of the log odds ratios for multitreatment studies
### see Gleser & Olkin (2009), equation (19.11), for the covariance equation
calc.v <- function(x) {
  v <- matrix(1/(x$xi.2[1]+1/2) + 1/(x$ni.2[1] - x$xi.2[1] + 1/2), nrow=nrow(x), ncol=nrow(x))
```

```
    diag(v) <- x$vi
    v
}
V <- bldiag(lapply(split(dat, dat$study), calc.v))

### add contrast matrix to dataset
dat <- contrmat(dat, grp1="trt.1", grp2="trt.2")
dat

### network meta-analysis using a contrast-based random-effects model
### by setting rho=1/2, tau^2 reflects the amount of heterogeneity for all treatment comparisons
res <- rma.mv(yi, V, mods = ~ self_help + ind_counseling + grp_counseling - 1,
              random = ~ comp | study, rho=1/2, data=dat)
res

### predicted odds ratios of interventions versus no contact
predict(res, newmods=diag(3), transf=exp, digits=2)

### fit random inconsistency effects model (see Law et al., 2016)
res <- rma.mv(yi, V, mods = ~ self_help + ind_counseling + grp_counseling - 1,
             random = list(~ comp | study, ~ comp | design), rho=1/2, phi=1/2, data=dat)
res
```

dat.hine1989 *Studies on Prophylactic Use of Lidocaine After a Heart Attack*

Description

Results from 6 studies evaluating mortality from prophylactic use of lidocaine in acute myocardial infarction.

Usage

```
dat.hine1989
```

Format

The data frame contains the following columns:

study	`numeric`	study number
source	`character`	source of data
n1i	`numeric`	number of patients in lidocaine group
n2i	`numeric`	number of patients in control group
ai	`numeric`	number of deaths in lidocaine group
ci	`numeric`	number of deaths in control group

Details

Hine et al. (1989) conducted a meta-analysis of death rates in randomized controlled trials in which prophylactic lidocaine was administered to patients with confirmed or suspected acute myocardial infarction. The dataset describes the mortality at the end of the assigned treatment period for control and intravenous lidocaine treatment groups for six studies. The question of interest is whether there is a detrimental effect of lidocaine. Because the studies were conducted to compare rates of arrhythmias following a heart attack, the studies, taken individually, are too small to detect important differences in mortality rates.

The data in this dataset were obtained from Table I in Normand (1999, p. 322).

Source

Normand, S. T. (1999). Meta-analysis: Formulating, evaluating, combining, and reporting. *Statistics in Medicine*, **18**(3), 321–359. https://doi.org/10.1002/(sici)1097-0258(19990215)18:3<321::aid-sim28>3.0.co;2-p

References

Hine, L. K., Laird, N., Hewitt, P., & Chalmers, T. C. (1989). Meta-analytic evidence against prophylactic use of lidocaine in acute myocardial infarction. *Archives of Internal Medicine*, **149**(12), 2694–2698. https://doi.org/10.1001/archinte.1989.00390120056011

Examples

```
### copy data into 'dat'
dat <- dat.hine1989

### calculate risk differences and corresponding sampling variances
dat <- escalc(measure="RD", n1i=n1i, n2i=n2i, ai=ai, ci=ci, data=dat)
dat

### meta-analysis of risk differences using a random-effects model
res <- rma(yi, vi, data=dat)
res
```

---

dat.ishak2007 *Studies on Deep-Brain Stimulation in Patients with Parkinson's disease*

Description

Results from 46 studies examining the effects of deep-brain stimulation on motor skills of patients with Parkinson's disease.

Usage

```
dat.ishak2007
```

Format

The data frame contains the following columns:

study	`character`	(first) author and year
y1i	`numeric`	observed mean difference at 3 months
v1i	`numeric`	sampling variance of the mean difference at 3 months
y2i	`numeric`	observed mean difference at 6 months
v2i	`numeric`	sampling variance of the mean difference at 6 months
y3i	`numeric`	observed mean difference at 12 months
v3i	`numeric`	sampling variance of the mean difference at 12 months
y4i	`numeric`	observed mean difference at the long-term follow-up
v4i	`numeric`	sampling variance of the mean difference at the long-term follow-up
mdur	`numeric`	mean disease duration (in years)
mbase	`numeric`	mean baseline UPDRS score

Details

Deep-brain stimulation (DBS), which is delivered through thin surgically implanted wires in specific areas of the brain and controlled by the patient, is meant to provide relief of the debilitating symptoms of Parkinson's disease. The dataset includes the results from 46 studies examining the effects of DBS of the subthalamic nucleus on motor functioning, measured with the Unified Parkinson's Disease Rating Scale (UPDRS). The effect size measure for this meta-analysis was the mean difference of the scores while the stimulator is active and the baseline scores (before implantation of the stimulator). Since lower scores on the UPDRS indicate better functioning, negative numbers indicate improvements in motor skills. Effects were generally measured at 3, 6, and 12 months after implantation of the stimulator, with some studies also including a further long-term follow-up. However, the number of measurements differed between studies - hence the missing data on some of the measurement occasions.

Since the same patients were followed over time within a study, effect size estimates from multiple measurement occasions are likely to be correlated. A multivariate model accounting for the correlation in the effects can be used to meta-analyze these data. A difficulty with this approach is the lack of information about the correlation of the measurements over time in the individual studies. The approach taken by Ishak et al. (2007) was to assume an autoregressive (AR1) structure for the estimates within the individual studies. In addition, the correlation in the true effects was modeled, again using an autoregressive structure.

Source

Ishak, K. J., Platt, R. W., Joseph, L., Hanley, J. A., & Caro, J. J. (2007). Meta-analysis of longitudinal studies. *Clinical Trials*, **4**(5), 525–539. https://doi.org/10.1177/1740774507083567

Examples

```
### copy data into 'dat' and examine data
dat <- dat.ishak2007
dat

### create long format dataset
dat.long <- reshape(dat, direction="long", idvar="study", v.names=c("yi","vi"),
                    varying=list(c(2,4,6,8), c(3,5,7,9)))
dat.long <- dat.long[order(dat.long$study, dat.long$time),]
rownames(dat.long) <- 1:nrow(dat.long)

### remove missing measurement occasions from dat.long
is.miss  <- is.na(dat.long$yi)
dat.long <- dat.long[!is.miss,]

### construct the full (block diagonal) V matrix with an AR(1) structure
rho.within <- .97 ### value as estimated by Ishak et al. (2007)
V <- lapply(split(with(dat, cbind(v1i, v2i, v3i, v4i)), dat$study), diag)
V <- lapply(V, function(v) sqrt(v) %*% toeplitz(ARMAacf(ar=rho.within, lag.max=3)) %*% sqrt(v))
V <- bldiag(V)
V <- V[!is.miss,!is.miss] ### remove missing measurement occasions from V

### plot data
with(dat.long, interaction.plot(time, study, yi, type="b", pch=19, lty="solid", xaxt="n",
               legend=FALSE, xlab="Time Point", ylab="Mean Difference", bty="l"))
axis(side=1, at=1:4, lab=c("1 (3 months)", "2 (6 months)", "3 (12 months)", "4 (12+ months)"))

### multivariate model with heteroscedastic AR(1) structure for the true effects
res <- rma.mv(yi, V, mods = ~ factor(time) - 1, random = ~ time | study,
              struct = "HAR", data = dat.long)
print(res, digits=2)
```

dat.kalaian1996 *Studies on the Effectiveness of Coaching for the SAT*

Description

Results from studies examining the effectiveness of coaching on the performance on the Scholastic Aptitude Test (SAT).

Usage

```
dat.kalaian1996
```

Format

The data frame contains the following columns:

id	`numeric`	row (effect) id
study	`character`	study identifier
year	`numeric`	publication year
n1i	`numeric`	number of participants in the coached group
n2i	`numeric`	number of participants in the uncoached group
outcome	`character`	subtest (verbal or math)
yi	`numeric`	standardized mean difference
vi	`numeric`	corresponding sampling variance

hrs	`numeric`	hours of coaching
ets	`numeric`	study conducted by the Educational Testing Service (ETS) (0 = no, 1 = yes)
homework	`numeric`	assignment of homework outside of the coaching course (0 = no, 1 = yes)
type	`numeric`	study type (1 = randomized study, 2 = matched study, 3 = nonequivalent comparison study)

Details

The effectiveness of coaching for the Scholastic Aptitude Test (SAT) has been examined in numerous studies. This dataset contains standardized mean differences comparing the performance of a coached versus uncoached group on the verbal and/or math subtest of the SAT. Studies may report a standardized mean difference for the verbal subtest, the math subtest, or both. In the latter case, the two standardized mean differences are not independent (since they were measured in the same group of subjects). The number of hours of coaching (variable `hrs`), whether the study was conducted by the Educational Testing Service (variable `ets`), whether homework was assigned outside of the coaching course (variable `homework`), and the study type (variable `type`) may be potential moderators of the treatment effect.

Note

The dataset was obtained from Table 1 in Kalaian and Raudenbush (1996). However, there appear to be some inconsistencies between the data in the table and those that were actually used for the analyses (see ‘Examples’).

Source

Kalaian, H. A., & Raudenbush, S. W. (1996). A multivariate mixed linear model for meta-analysis. *Psychological Methods*, **1**(3), 227–235. https://doi.org/10.1037/1082-989X.1.3.227

Examples

```
### copy data into 'dat'
dat <- dat.kalaian1996

### check ranges
range(dat$yi[dat$outcome == "verbal"]) # -0.35 to 0.74 according to page 230
range(dat$yi[dat$outcome == "math"])   # -0.53 to 0.60 according to page 231

### comparing this with Figure 1 in the paper reveals some discrepancies
par(mfrow=c(1,2), mar=c(5,4,1,1))
plot(log(dat$hrs[dat$outcome == "verbal"]), dat$yi[dat$outcome == "verbal"],
     pch=19, xlab="Log(Coaching Hours)", ylab="Effect Size (verbal)",
     xlim=c(1,6), ylim=c(-0.5,1), xaxs="i", yaxs="i")
abline(h=c(-0.5,0,0.5), lty="dotted")
abline(v=log(c(5,18)),  lty="dotted")
plot(log(dat$hrs[dat$outcome == "math"]), dat$yi[dat$outcome == "math"],
     pch=19, xlab="Log(Coaching Hours)", ylab="Effect Size (math)",
     xlim=c(1,6), ylim=c(-1.0,1), xaxs="i", yaxs="i")
abline(h=c(-0.5,0,0.5), lty="dotted")
abline(v=log(c(5,18)),  lty="dotted")

### construct variance-covariance matrices assuming rho = 0.66
```

```
vcalc <- function(v, rho) {
   S <- diag(sqrt(v), nrow=length(v), ncol=length(v))
   R <- matrix(rho, nrow=length(v), ncol=length(v))
   diag(R) <- 1
   S %*% R %*% S
}
V <- lapply(split(dat$v, dat$study), vcalc, rho=0.66)
V <- bldiag(V, order=dat$study)

### fit multivariate random-effects model
res <- rma.mv(yi, V, mods = ~ outcome - 1,
              random = ~ outcome | study, struct="UN",
              data=dat, digits=3)
res

### test whether the effect differs for the math and verbal subtest
anova(res, X=c(1,-1))

### log-transform and mean center the hours of coaching variable
dat$loghrs <- log(dat$hrs) - mean(log(dat$hrs), na.rm=TRUE)

### fit multivariate model with log(hrs) as moderator
res <- rma.mv(yi, V, mods = ~ outcome + outcome:loghrs - 1,
              random = ~ outcome | study, struct="UN",
              data=dat, digits=3)
res

### fit model with tau2 = 0 for outcome verbal (which also constrains rho = 0)
res <- rma.mv(yi, V, mods = ~ outcome + outcome:loghrs - 1,
              random = ~ outcome | study, struct="UN", tau2=c(NA,0),
              data=dat, digits=3)
res
```

---

dat.kearon1998 *Studies on the Accuracy of Venous Ultrasonography for the Diagonosis of Deep Venous Thrombosis*

Description

Results from diagnostic accuracy studies examining the accuracy of venous ultrasonography for the diagonosis of deep venous thrombosis.

Usage

```
dat.kearon1998
```

Format

The data frame contains the following columns:

id	`numeric`	study id
author	`character`	study author(s)
year	`numeric`	publication year
patients	`character`	patient group (either symptomatic or asymptomatic patients)
tp	`numeric`	number of true positives
np	`numeric`	number of positive patients (cases)
tn	`numeric`	number of true negatives
nn	`numeric`	number of negative patients (non-cases)

Details

The studies included in the dataset examined the accuracy of venous ultrasonography for the diagnossis of a first deep venous thrombosis in symptomatic and asymptomatic patients. Cases and non-cases were determined based on contrast venography. Venous ultrasonography was then used to make a diagnosis, leading to a given number of true positives and negatives.

A subset of this dataset (using only the studies with asymptomatic patients) was used by Deeks et al. (2005) to illustrate methods for detecting publication bias (or small-study effects) in meta-analyses of diagnostic accuracy studies.

Source

Kearon, C., Julian, J. A., Math, M., Newman, T. E., & Ginsberg, J. S. (1998). Noninvasive diagnosis of deep venous thrombosis. *Annals of Internal Medicine*, **128**(8), 663–677. https://doi.org/10.7326/0003-4819-128-8-199804150-00011

References

Deeks, J. J., Macaskill, P., & Irwig, L. (2005). The performance of tests of publication bias and other sample size effects in systematic reviews of diagnostic test accuracy was assessed. *Journal of Clinical Epidemiology*, **58**(9), 882–893. https://doi.org/10.1016/j.jclinepi.2005.01.016

Examples

```
### copy data into 'dat'
dat <- dat.kearon1998

### calculate diagnostic log odds ratios and corresponding sampling variances
dat <- escalc(measure="OR", ai=tp, n1i=np, ci=nn-tn, n2i=nn, data=dat, add=1/2, to="all")
dat

### fit random-effects model for the symptomatic patients
res <- rma(yi, vi, data=dat, subset=patients=="symptomatic")
res

### fit random-effects model for the asymptomatic patients
res <- rma(yi, vi, data=dat, subset=patients=="asymptomatic")
res

### estimated average diagnostic odds ratio (with 95% CI)
predict(res, transf=exp, digits=2)
```

```
### regression test for funnel plot asymmetry using SE as predictor
reg <- regtest(res, model=”lm”)
reg

### corresponding funnel plot
funnel(res, atransf=exp, xlim=c(0,7), at=log(c(1,10,100,1000)), ylim=c(0,1.5), steps=4)
ys <- seq(0, 2, length=100)
lines(coef(reg$fit)[1] + coef(reg$fit)[2]*ys, ys, lwd=2, lty=3)

### regression test for funnel plot asymmetry using total sample size as predictor
reg <- regtest(res, model=”lm”, predictor=”ni”)
reg

### corresponding funnel plot
funnel(res, yaxis=”ni”, atransf=exp, xlim=c(0,7), at=log(c(1,10,100,1000)), ylim=c(0,300), steps=4)
ys <- seq(0, 300, length=100)
lines(coef(reg$fit)[1] + coef(reg$fit)[2]*ys, ys, lwd=2, lty=3)

### regression test for funnel plot asymmetry using 1/sqrt(ESS) as predictor (Deeks et al., 2005)
dat$invessi <- 1/(4*dat$np) + 1/(4*dat$nn)
tmp <- rma(yi, invessi, data=dat, subset=patients==”asymptomatic”)
reg <- regtest(tmp, model=”lm”)
reg

### corresponding funnel plot
funnel(tmp, atransf=exp, xlim=c(0,7), at=log(c(1,10,100,1000)), ylim=c(0,.15), steps=4,
       refline=coef(res), level=0, ylab=”1/root(ess)”)
ys <- seq(0, .20, length=100)
lines(coef(reg$fit)[1] + coef(reg$fit)[2]*ys, ys, lwd=2, lty=3)

### convert data to long format
dat <- to.long(measure=”OR”, ai=tp, n1i=np, ci=tn, n2i=nn,
               data=dat.kearon1998, subset=patients==”asymptomatic”)
dat <- dat[9:12]
levels(dat$group) <- c(”sensitivity”, ”specificity”)
dat

### calculate logit-transformed sensitivities
dat <- escalc(measure=”PLO”, xi=out1, mi=out2, data=dat, add=1/2, to=”all”,
              include=group==”sensitivity”)
dat

### calculate logit-transformed specificities
dat <- escalc(measure=”PLO”, xi=out1, mi=out2, data=dat, add=1/2, to=”all”,
              include=group==”specificity”)
dat

### bivariate random-effects model for logit sensitivity and specificity
res <- rma.mv(yi, vi, mods = ~ group - 1, random = ~ group | study, struct=”UN”, data=dat)
res

### estimated average sensitivity and specificity based on the model
```

```
predict(res, newmods = rbind(c(1,0),c(0,1)), transf=transf.ilogit, tau2.levels=c(1,2), digits=2)

### estimated average diagnostic odds ratio based on the model
predict(res, newmods = c(1,1), transf=exp, digits=2)
```

dat.knapp2017 — *Studies on Differences in Planning Performance in Schizophrenia Patients versus Healthy Controls*

Description

Results from 31 studies examining differences in planning performance in schizophrenia patients versus healthy controls.

Usage

```
dat.knapp2017
```

Format

The data frame contains the following columns:

author	`character`	study author(s)
year	`numeric`	publication year
study	`numeric`	study id number
yi	`numeric`	standardized mean difference for planning performance
vi	`numeric`	corresponding sampling variance
difficulty	`numeric`	task difficulty
n_sz	`numeric`	number of schizophrenic patients
n_hc	`numeric`	number of healthy controls
comp	`numeric`	id for comparisons within studies
yi	`numeric`	standardized mean difference for IQ
vi	`numeric`	corresponding sampling variance

Details

The studies included in this dataset examined differences between schizophrenia patients and healthy controls with respect to their performance on the tower of London test (`https://en.wikipedia.org/wiki/Tower_of_London_test`) or a similar cognitive tasks measuring planning ability. The outcome measure for this meta-analysis was the standardized mean difference (with positive values indicating better performance in the healthy controls compared to the schizophrenia patients).

The dataset has a more complex structure for two reasons:

1. Studies 2, 3, 9, and 20 included more than schizophrenia patient group and the standardized mean differences were computed by comparing these groups against a single healthy control group.
2. Studies 6, 12, 14, 15, 18, 19, 22, and 26 had the patients and controls complete different tasks of varying complexity (essentially the average number of moves required to complete a task).

Both of these issues lead to correlated sampling errors, which should be taken into consideration in the analysis.

Source

Knapp, F., Viechtbauer, W., Leonhart, R., Nitschke, K., & Kaller, C. P. (2017). Planning performance in schizophrenia patients: A meta-analysis of the influence of task difficulty and clinical and sociodemographic variables. *Psychological Medicine*, **47**(11), 2002–2016. https://doi.org/10.1017/S0033291717000459

Examples

```
### copy data into 'dat' and examine data
dat <- dat.knapp2017
dat

### fit a standard random-effects model ignoring correlated sampling errors
res <- rma(yi, vi, data=dat)
res

### fit a multilevel model with random effects for studies and comparisons within studies
res <- rma.mv(yi, vi, random = ~ 1 | study/comp, data=dat)
res

### construct an approximate V matrix assuming a correlation of 0.4 for the sampling errors
### of different comparisons within the same study
V <- lapply(split(dat$vi, dat$study), function(v) {
   S <- diag(sqrt(v), nrow=length(v), ncol=length(v))
   R <- matrix(0.4, nrow=length(v), ncol=length(v))
   diag(R) <- 1
   S %*% R %*% S
})
V <- bldiag(V, order=dat$study)

### fit the same multilevel model, but now use this V matrix in the model
res <- rma.mv(yi, V, random = ~ 1 | study/comp, data=dat)
res

### use cluster-robust inference methods based on this model
robust(res, cluster=dat$study)

### examine if task difficulty is a potential moderator of the effect
res <- rma.mv(yi, V, mods = ~ difficulty, random = ~ 1 | study/comp, data=dat)
res
sav <- robust(res, cluster=dat$study)
sav

### draw bubble plot
regplot(sav, xlab="Task Difficulty", ylab="Standardized Mean Difference", las=1, digits=1, bty="l")
```

dat.konstantopoulos2011 *Studies on the Effects of Modified School Calendars on Student Achievement*

Description

Results from 56 studies on the effects of modified school calendars on student achievement.

Usage

```
dat.konstantopoulos2011
```

Format

The data frame contains the following columns:

district	`numeric`	district id number
school	`numeric`	school id number (within district)
study	`numeric`	study id number
yi	`numeric`	standardized mean difference
vi	`numeric`	corresponding sampling variance
year	`numeric`	year of the study

Details

Instead of following the more traditional school calendar with a long summer break (in addition to a short winter and spring break), some schools have switched to a modified school calendar comprising more frequent but shorter intermittent breaks (e.g., 9 weeks of school followed by 3 weeks off), while keeping the total number of days at school approximately the same. The effects of using such a modified calendar on student achievement have been examined in a number of studies and were meta-analyzed by Cooper et al. (2003).

The dataset (taken from Konstantopoulos, 2011) contains the results from 56 studies, each comparing the level of academic achievement in a group of students following a modified school calendar with that of a group of students following a more traditional school calendar. The difference between the two groups was quantified in terms of a standardized mean difference (with positive values indicating a higher mean level of achievement in the group following the modified school calendar).

The studies were conducted at various schools that were clustered within districts. The data therefore have a multilevel structure, with schools nested within districts. A multilevel meta-analysis of these data can be used to estimate and account for the amount of heterogeneity between districts and between schools within districts.

Source

Konstantopoulos, S. (2011). Fixed effects and variance components estimation in three-level meta-analysis. *Research Synthesis Methods*, **2**(1), 61–76. https://doi.org/10.1002/jrsm.35

References

Cooper, H., Valentine, J. C., Charlton, K., & Melson, A. (2003). The effects of modified school calendars on student achievement and on school and community attitudes. *Review of Educational Research*, **73**(1), 1–52. https://doi.org/10.3102/00346543073001001

Examples

```
### copy data into 'dat' and examine data
dat <- dat.konstantopoulos2011
dat

### regular random-effects model
res <- rma(yi, vi, data=dat)
print(res, digits=3)

### regular random-effects model using rma.mv()
res <- rma.mv(yi, vi, random = ~ 1 | study, data=dat)
print(res, digits=3)

### multilevel random-effects model
res.ml <- rma.mv(yi, vi, random = ~ 1 | district/school, data=dat)
print(res.ml, digits=3)

### profile variance components
profile(res.ml, progbar=FALSE)

### multivariate parameterization of the model
res.mv <- rma.mv(yi, vi, random = ~ factor(school) | district, data=dat)
print(res.mv, digits=3)

### tau^2 from multivariate model = sum of the two variance components from the multilevel model
round(sum(res.ml$sigma2), 3)

### rho from multivariate model = intraclass correlation coefficient based on the multilevel model
round(res.ml$sigma2[1] / sum(res.ml$sigma2), 3)
```

dat.landenberger2005 *Studies on the Effectiveness of CBT for Reducing Recidivism*

Description

Results from 58 studies on the effectiveness of cognitive-behavioral therapy (CBT) for reducing recidivism in juvenile and adult offenders.

Usage

```
dat.landenberger2005
```

Format

The data frame contains the following columns:

study	`character`	(first) author and year
pubtype	`character`	publication type (book chapter, journal article, report, or thesis)
country	`character`	country where study was carried out (Canada, New Zealand, UK, or USA)
design	`character`	study design (matched groups, nonequivalent groups, or randomized trial)
program	`character`	purpose of setting up the CBT program (for demonstration, practice, or research purposes)
setting	`character`	treatment setting (community or prison)
designprob	`character`	indication of study design problems (no, favors the control group, or favors the treatment gr
n.ctrl.rec	`numeric`	number of recidivists in the control group
n.ctrl.non	`numeric`	number of non-recidivists in the control group
n.cbt.rec	`numeric`	number of recidivists in the CBT group
n.cbt.non	`numeric`	number of non-recidivists in the CBT group
interval	`numeric`	recidivism interval (in months)
group	`numeric`	study group (adults or juveniles)
age	`numeric`	mean age of the study group
male	`numeric`	percentage of males in the study group
minority	`numeric`	percentage of minorities in the study group
length	`numeric`	treatment length (in weeks)
sessions	`numeric`	number of CBT sessions per week
hrs_week	`numeric`	treatment hours per week
hrs_total	`numeric`	total hours of treatment
cbt.cogskills	`character`	CBT component: cognitive skills (yes, no)
cbt.cogrestruct	`character`	CBT component: cognitive restructuring (yes, no)
cbt.intpprbsolv	`character`	CBT component: interpersonal problem solving (yes, no)
cbt.socskills	`character`	CBT component: social skills (yes, no)
cbt.angerctrl	`character`	CBT component: anger control (yes, no)
cbt.victimimpact	`character`	CBT component: victim impact (yes, no)
cbt.subabuse	`character`	CBT component: substance abuse (yes, no)
cbt.behavmod	`character`	CBT component: behavior modification (yes, no)
cbt.relapseprev	`character`	CBT component: relapse prevention (yes, no)
cbt.moralrsng	`character`	CBT component: moral reasoning (yes, no)
cbt.roletaking	`character`	CBT component: role taking (yes, no)
cbt.other	`character`	CBT component: other (yes, no)

Details

Landenberger and Lipsey (2005) conducted a meta-analysis of 58 experimental and quasi-experimental studies of the effects of cognitive-behavioral therapy (CBT) on the recidivism rates of adult and juvenile offenders (see also Lipsey et al., 2007). The present dataset includes the results of these studies and a range of potential moderator variables to identify factors associated with variation in treatment effects.

Source

Personal communication.

References

Landenberger, N. A., & Lipsey, M. W. (2005). The positive effects of cognitive-behavioral programs for offenders: A meta-analysis of factors associated with effective treatment. *Journal of*

Experimental Criminology, **1**, 451–476. https://doi.org/10.1007/s11292-005-3541-7

Lipsey, M. W., Landenberger, N. A., & Wilson, S. J. (2007). Effects of cognitive-behavioral programs for criminal offenders. *Campbell Systematic Reviews*, **3**(1), 1–27. https://doi.org/10.4073/csr.2007.6

Examples

```
### copy data into 'dat' and examine data
dat <- dat.landenberger2005
head(dat)

### calculate log odds ratios (for non-recidivism in CBT vs. control groups) and sampling variances
dat <- escalc(measure="OR", ai=n.cbt.non, bi=n.cbt.rec, ci=n.ctrl.non, di=n.ctrl.rec, data=dat)

### fit random-effects model
res <- rma(yi, vi, data=dat)
res

### estimated average OR and corresponding 95% CI/PI
predict(res, transf=exp, digits=2)

### examine if number of treatment sessions per week is a potential moderator
res <- rma(yi, vi, mods = ~ sessions, data=dat)
res

### predicted ORs for 1, 2, 5, or 10 sessions per week
predict(res, newmods=c(1,2,5,10), transf=exp, digits=2)
```

dat.laopaiboon2015 *Studies on the Effectiveness of Azithromycin for Treating Lower Respiratory Tract Infections*

Description

Results from 15 studies on the effectiveness of azithromycin versus amoxycillin or amoxycillin/clavulanic acid (amoxyclav) in the treatment of acute lower respiratory tract infections.

Usage

```
dat.laopaiboon2015
```

Format

The data frame contains the following columns:

author	`character`	author(s)
year	`numeric`	publication year
ai	`numeric`	number of clinical failures in the group treated with azithromycin
n1i	`numeric`	number of patients in the group treated with azithromycin
ci	`numeric`	number of clinical failures in the group treated with amoxycillin or amoxyclav

n2i	`numeric`	number of patients in the group treated with amoxycillin or amoxyclav
age	`character`	whether the trial included adults or children
diag.ab	`numeric`	trial included patients with a diagnosis of acute bacterial bronchitis
diag.cb	`numeric`	trial included patients with a diagnosis of chronic bronchitis with acute exacerbation
diag.pn	`numeric`	trial included patients with a diagnosis of pneumonia
ctrl	`character`	antibiotic in control group (amoxycillin or amoxyclav)

Details

Azithromycin is an antibiotic useful for the treatment of a number of bacterial infections. Laopaiboon et al. (2015) conducted a meta-analysis of trials comparing the effectiveness of azithromycin versus amoxycillin or amoxycillin/clavulanic acid (amoxyclav) in the treatment of acute lower respiratory tract infections, including acute bacterial bronchitis, acute exacerbations of chronic bronchitis, and pneumonia. The results from 15 trials are included in this dataset.

Source

Laopaiboon, M., Panpanich, R., & Swa Mya, K. (2015). Azithromycin for acute lower respiratory tract infections. *Cochrane Database of Systematic Reviews*, **3**, CD001954. https://doi.org/10.1002/14651858.CD001954.pub4

Examples

```
### copy data into 'dat' and examine data
dat <- dat.laopaiboon2015
dat

### analysis using the Mantel-Haenszel method
rma.mh(measure="RR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, digits=3)

### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat)

### random-effects model
res <- rma(yi, vi, data=dat)
res

### average risk ratio with 95% CI
predict(res, transf=exp)
```

`dat.lau1992` *Studies on Intravenous Streptokinase for Acute Myocardial Infarction*

Description

Results from 33 trials comparing intravenous streptokinase versus placebo or no therapy in patients who had been hospitalized for acute myocardial infarction.

Usage

```
dat.lau1992
```

Format

The data frame contains the following columns:

trial	`character`	trial name
year	`numeric`	publication year
ai	`numeric`	number of deaths in the streptokinase group
n1i	`numeric`	number of patients in the streptokinase group
ci	`numeric`	number of deaths in the control group
n2i	`numeric`	number of patients in the control group

Details

In the paper by Lau et al. (1992), the data are used to illustrate the idea of a cumulative meta-analysis, where the results are updated as each trial is added to the dataset. See ‘Examples’ for code that replicates the results and shows corresponding forest plots.

Source

Lau, J., Antman, E. M., Jimenez-Silva, J., Kupelnick, B., Mosteller, F., & Chalmers, T. C. (1992). Cumulative meta-analysis of therapeutic trials for myocardial infarction. *New England Journal of Medicine*, **327**(4), 248–254. https://doi.org/10.1056/NEJM199207233270406

Examples

```
### copy data into 'dat' and examine data
dat <- dat.lau1992
dat

### meta-analysis of log odds ratios using the MH method
res <- rma.mh(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, slab=trial)
print(res, digits=2)

### forest plot
forest(res, xlim=c(-10,9), atransf=exp, at=log(c(.01, 0.1, 1, 10, 100)),
       header=TRUE, top=2, ilab=dat$year, ilab.xpos=-6)
text(-6, 35, "Year", font=2)

### cumulative meta-analysis
sav <- cumul(res)

### forest plot of the cumulative results
forest(sav, xlim=c(-5,4), atransf=exp, at=log(c(0.1, 0.5, 1, 2, 10)),
       header=TRUE, top=2, ilab=dat$year, ilab.xpos=-3)
text(-3, 35, "Year", font=2)
id <- c(4, 8, 15, 33) # rows for which the z/p-values should be shown (as in Lau et al., 1992)
text(1.1, (res$k:1)[id], paste0("z = ", formatC(sav$zval[id], format="f", digits=2),
                                ", p = ", formatC(sav$pval[id], format="f", digits=4)))
```

dat.lee2004 — *Studies on Acupoint P6 Stimulation for Preventing Nausea*

Description

Results from studies examining the effectiveness of wrist acupuncture point P6 stimulation for preventing postoperative nausea.

Usage

```
dat.lee2004
```

Format

The data frame contains the following columns:

id	`numeric`	trial id number
study	`character`	first author
year	`numeric`	study year
ai	`numeric`	number of patients experiencing nausea in the treatment group
n1i	`numeric`	total number of patients in treatment group
ci	`numeric`	number of patients experiencing nausea in the sham group
n2i	`numeric`	total number of patients in the sham group

Details

Postoperative nausea and vomiting are common complications following surgery and anaesthesia. As an alternative to drug therapy, acupuncture has been studied as a potential treatment in several trials. The dataset contains the results from 16 clinical trials examining the effectiveness of wrist acupuncture point P6 stimulation for preventing postoperative nausea.

Source

Lee, A., & Done, M. L. (2004). Stimulation of the wrist acupuncture point P6 for preventing postoperative nausea and vomiting. *Cochrane Database of Systematic Reviews*, **3**, CD003281. https://doi.org/10.1002/14651858.CD003281.pub2

Examples

```
### copy data into 'dat' and examine data
dat <- dat.lee2004
dat

### meta-analysis based on log risk ratios
res <- rma(measure="RR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat)
res
predict(res, transf=exp, digits=2)
```

dat.li2007 *Studies on the Effectiveness of Intravenous Magnesium in Acute Myocardial Infarction*

Description

Results from 22 trials examining the effectiveness of intravenous magnesium in the prevention of death following acute myocardial infarction.

Usage

```
dat.li2007
```

Format

The data frame contains the following columns:

id	`numeric`	trial id number
study	`character`	first author or trial name
year	`numeric`	publication year
ai	`numeric`	number of deaths in the magnesium group
n1i	`numeric`	number of patients in the magnesium group
ci	`numeric`	number of deaths in the control group
n2i	`numeric`	number of patients in the control group

Details

The dataset includes the results from 22 randomized clinical trials that examined the effectiveness of intravenous magnesium in the prevention of death following acute myocardial infarction. It is similar to the dataset `dat.egger2001`, with some slight differences in the included trials and data used.

Source

Li, J., Zhang, Q., Zhang, M., & Egger, M. (2007). Intravenous magnesium for acute myocardial infarction. *Cochrane Database of Systematic Reviews*, **2**, CD002755. https://doi.org/10.1002/14651858.CD002755.pub2

See Also

`dat.egger2001`

Examples

```
### copy data into 'dat' and examine data
dat <- dat.li2007
dat

### meta-analysis of all trials except ISIS-4
```

```
res <- rma(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, method="FE", subset=-14)
print(res, digits=2)
predict(res, transf=exp, digits=2)

### meta-analysis of all trials including ISIS-4
res <- rma(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, method="FE")
print(res, digits=2)
predict(res, transf=exp, digits=2)

### contour-enhanced funnel plot centered at 0
funnel(res, refline=0, level=c(90, 95, 99), shade=c("white", "gray", "darkgray"))
```

`dat.lim2014` *Studies on the association between maternal size, offspring size, and number of offsprings*

Description

Results from studies examining the association between maternal size, offspring size, and number of offsprings.

Usage

```
dat.lim2014
```

Format

The object is a list containing data frames `m_o_size`, `m_o_fecundity`, `o_o_unadj`, and `o_o_adj` that contain the following columns and the corresponding phylogenetic trees called `m_o_size_tree`, `m_o_fecundity_tree`, `o_o_unadj_tree`, and `o_o_adj_tree`:

article	`numeric`	article id
author	`character`	study author(s)
year	`numeric`	publication year
species	`character`	species
amniotes	`character`	whether the species was amniotic
environment	`character`	whether the species were wild or captive
reprounit	`character`	whether the data were based on lifetime reproductive output or a single reproductive event (only
ri	`numeric`	correlation coefficient
ni	`numeric`	sample size

Details

The object `dat.lim2014` includes 4 datasets:

`m_o_size`	on the correlation between maternal size and offspring size
`m_o_fecundity`	on the correlation between maternal size and number of offsprings
`o_o_unadj`	on the correlation between offspring size and number of offsprings
`o_o_adj`	on the correlation between offspring size and number of offsprings adjusted for maternal size

Objects `m_o_size_tree`, `m_o_fecundity_tree`, `o_o_unadj_tree`, and `o_o_adj_tree` are the corresponding phylogenetic trees for the species included in each of these datasets.

Source

Lim, J. N., Senior, A. M., & Nakagawa, S. (2014). Heterogeneity in individual quality and reproductive trade-offs within species. *Evolution*, **68**(8), 2306–2318. https://doi.org/10.1111/evo.12446

References

Hadfield, J. D., & Nakagawa, S. (2010). General quantitative genetic methods for comparative biology: Phylogenies, taxonomies and multi-trait models for continuous and categorical characters. *Journal of Evolutionary Biology*, **23**(3), 494–508. https://doi.org/10.1111/j.1420-9101.2009.01915.x

Nakagawa, S., & Santos, E. S. A. (2012). Methodological issues and advances in biological meta-analysis. *Evolutionary Ecology*, **26**(5), 1253–1274. https://doi.org/10.1007/s10682-012-9555-5

Examples

```
### copy data into 'dat' and examine data
dat <- dat.lim2014$o_o_unadj
head(dat)

### calculate r-to-z transformed correlations and corresponding sampling variances
dat <- escalc(measure="ZCOR", ri=ri, ni=ni, data=dat)

## Not run:
### load 'ape' package
require(ape)

### copy tree to 'tree'
tree <- dat.lim2014$o_o_unadj_tree

### compute branch lengths
tree <- compute.brlen(tree)

### compute phylogenetic correlation matrix
A <- vcv(tree, corr=TRUE)

### make copy of the species variable
dat$species.phy <- dat$species

### create effect size id variable
dat$esid <- 1:nrow(dat)

### fit multilevel phylogenetic meta-analytic model
res <- rma.mv(yi, vi,
   random = list(~ 1 | article, ~ 1 | esid, ~ 1 | species, ~ 1 | species.phy),
   R=list(species.phy=A), data=dat)
res

## End(Not run)
```

dat.linde2005 *Studies on the Effectiveness of St. John's Wort for Treating Depression*

Description

Results from 26 studies on the effectiveness of Hypericum perforatum extracts (St. John's wort) for treating depression.

Usage

```
dat.linde2005
```

Format

The data frame contains the following columns:

id	`numeric`	study number
study	`character`	study author(s)
year	`numeric`	publication year
country	`character`	study location
ni	`numeric`	total sample size
major	`numeric`	sample restricted to patients who met criteria for major depression
baseline	`numeric`	HRSD baseline score
version	`numeric`	HRSD version (17 or 21 items)
duration	`numeric`	study duration (in weeks)
prep	`character`	Hypericum extract preparation
dosage	`numeric`	dosage (in mg)
response	`numeric`	definition of response (see 'Details')
ai	`numeric`	number of responses in treatment group
n1i	`numeric`	number of patients in treatment group
ci	`numeric`	number of responses in placebo group
n2i	`numeric`	number of patients in placebo group
group	`numeric`	stratification variable used by the authors (see 'Details')

Details

The dataset includes the results from 26 double-blind placebo-controlled trials on the effectiveness of Hypericum perforatum extracts (St. John's wort) for treating depression (note that 2 studies did not provide sufficient response information).

Data were extracted from Table 1 and Figure 3 from Linde et al. (2005). For study duration, the assessment week (instead of the total study duration) was coded for Philipp et al. (1999) and Montgomery et al. (2000). For dosage, the midpoint was coded when a range of values was given.

The definition of what constitutes a `response` differed across studies and is coded as follows:

1. HRSD score reduction of at least 50% or HRSD score after therapy <10,
2. HRSD reduction of at least 50%,

3. based on HRSD scale but exact definition not reported,
4. global patient assessment of efficacy,
5. at least ‘much improved’ on the Clinical Global Impression sub-scale for global improvement.

The group variable corresponds to the variable used by Linde et al. (2005) to stratify their analyses and is coded as follows:

1. smaller trials restricted to major depression,
2. larger trials restricted to major depression,
3. smaller trials not restricted to major depression,
4. larger trials not restricted to major depression.

Source

Linde, K., Berner, M., Egger, M., & Mulrow, C. (2005). St John’s wort for depression: Meta-analysis of randomised controlled trials. *British Journal of Psychiatry*, **186**(2), 99–107. https://doi.org/10.1192/bjp.186.2.99

References

Viechtbauer, W. (2007). Accounting for heterogeneity via random-effects models and moderator analyses in meta-analysis. *Zeitschrift für Psychologie / Journal of Psychology*, **215**(2), 104–121. https://doi.org/10.1027/0044-3409.215.2.104

Examples

```
### copy data into 'dat'
dat <- dat.linde2005

### remove studies with no response information and study with no responses in either group
dat <- dat[-c(5,6,26),]

### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=ai, ci=ci, n1i=n1i, n2i=n2i, data=dat)
dat

### meta-analysis of the log risk ratios using a random-effects model
res <- rma(yi, vi, data=dat, method="DL")
res

### mixed-effects meta-regression model with stratification variable
res <- rma(yi, vi, mods = ~ factor(group) - 1, data=dat, method="DL")
res

### predicted average risk ratio for each level of the stratification variable
predict(res, newmods=diag(4), transf=exp, digits=2)
```

`dat.lopez2019` *Studies on the Effectiveness of CBT for Depression*

Description

Results from 76 studies examining the effectiveness of cognitive behavioral therapy (CBT) for depression in adults.

Usage

```
dat.lopez2019
```

Format

The data frame contains the following columns:

study	`character`	(first) author and year of study
treatment	`character`	treatment provided (see 'Details')
scale	`character`	scale used to measure depression symptoms
n	`numeric`	group size
diff	`numeric`	standardized mean change
se	`numeric`	corresponding standard error
group	`numeric`	type of therapy (0 = individual, 1 = group therapy)
tailored	`numeric`	whether the intervention was tailored to each patient (0 = no, 1 = yes)
sessions	`numeric`	number of sessions
length	`numeric`	average session length (in minutes)
intensity	`numeric`	product of sessions and length
multi	`numeric`	intervention included multimedia elements (0 = no, 1 = yes)
cog	`numeric`	intervention included cognitive techniques (0 = no, 1 = yes)
ba	`numeric`	intervention included behavioral activation (0 = no, 1 = yes)
psed	`numeric`	intervention included psychoeducation (0 = no, 1 = yes)
home	`numeric`	intervention included homework (0 = no, 1 = yes)
prob	`numeric`	intervention included problem solving (0 = no, 1 = yes)
soc	`numeric`	intervention included social skills training (0 = no, 1 = yes)
relax	`numeric`	intervention included relaxation (0 = no, 1 = yes)
goal	`numeric`	intervention included goal setting (0 = no, 1 = yes)
final	`numeric`	intervention included a final session (0 = no, 1 = yes)
mind	`numeric`	intervention included mindfulness (0 = no, 1 = yes)
act	`numeric`	intervention included acceptance and commitment therapy (0 = no, 1 = yes)

Details

The dataset includes the results from 76 studies examining the effectiveness of cognitive behavioral therapy (CBT) for treating depression in adults. Studies included two or more of the following treatments/conditions:

1. treatment as usual (TAU),

2. no treatment,
3. wait list,
4. psychological or attention placebo,
5. face-to-face CBT,
6. multimedia CBT,
7. hybrid CBT (i.e., multimedia CBT with one or more face-to-face sessions).

Multimedia CBT was defined as CBT delivered via self-help books, audio/video recordings, telephone, computer programs, apps, e-mail, or text messages.

Variable diff is the standardized mean change within each group, with negative values indicating a decrease in depression symptoms.

Source

Personal communication.

References

López-López, J. A., Davies, S. R., Caldwell, D. M., Churchill, R., Peters, T. J., Tallon, D., Dawson, S., Wu, Q., Li, J., Taylor, A., Lewis, G., Kessler, D. S., Wiles, N., & Welton, N. J. (2019). The process and delivery of CBT for depression in adults: A systematic review and network meta-analysis. *Psychological Medicine*, **49**(12), 1937–1947. https://doi.org/10.1017/S003329171900120X

Examples

```
### copy data into 'dat' and examine data
dat <- dat.lopez2019
dat[1:10,1:6]

### create network graph ('igraph' package must be installed)
## Not run:
require(igraph)
pairs <- data.frame(do.call(rbind,
  sapply(split(dat$treatment, dat$study), function(x) t(combn(x,2)))), stringsAsFactors=FALSE)
pairs$X1 <- factor(pairs$X1, levels=sort(unique(dat$treatment)))
pairs$X2 <- factor(pairs$X2, levels=sort(unique(dat$treatment)))
tab <- table(pairs[,1], pairs[,2])
tab # adjacency matrix
g <- graph_from_adjacency_matrix(tab, mode = "plus", weighted=TRUE, diag=FALSE)
plot(g, edge.curved=FALSE, edge.width=E(g)$weight/2,
   layout=layout_in_circle(g, order=c("Wait list", "No treatment", "TAU", "Multimedia CBT",
                                      "Hybrid CBT", "F2F CBT", "Placebo")),
   vertex.size=45, vertex.color="lightgray", vertex.label.color="black", vertex.label.font=2)
## End(Not run)

### restructure data into wide format
dat <- to.wide(dat, study="study", grp="treatment", ref="TAU",
               grpvars=c("diff","se","n"), postfix=c("1","2"))

### compute contrasts between treatment pairs and corresponding sampling variances
```

```
dat$yi <- with(dat, diff1 - diff2)
dat$vi <- with(dat, se1^2 + se2^2)

### calculate the variance-covariance matrix for multitreatment studies
calc.v <- function(x) {
   v <- matrix(x$se2[1]^2, nrow=nrow(x), ncol=nrow(x))
   diag(v) <- x$vi
   v
}
V <- bldiag(lapply(split(dat, dat$study), calc.v))

### add contrast matrix to the dataset
dat <- contrmat(dat, grp1="treatment1", grp2="treatment2")

### network meta-analysis using a contrast-based random-effects model
### by setting rho=1/2, tau^2 reflects the amount of heterogeneity for all treatment comparisons
### the treatment left out (TAU) becomes the reference level for the treatment comparisons
res <- rma.mv(yi, V, data=dat,
      mods = ~ No.treatment + Wait.list + Placebo + F2F.CBT + Hybrid.CBT + Multimedia.CBT - 1,
         random = ~ comp | study, rho=1/2)
res

### forest plot of the contrast estimates (treatments versus TAU)
forest(coef(res), diag(vcov(res)), slab=sub(".", " ", names(coef(res)), fixed=TRUE),
       xlim=c(-5,5), alim=c(-3,3), psize=1, header="Treatment",
       xlab="Difference in Standardized Mean Change (compared to TAU)")

### fit random inconsistency effects model
res <- rma.mv(yi, V, data=dat,
      mods = ~ No.treatment + Wait.list + Placebo + F2F.CBT + Hybrid.CBT + Multimedia.CBT - 1,
         random = list(~ comp | study, ~ comp | design), rho=1/2, phi=1/2)
res
```

dat.maire2019 *Studies on Temporal Trends in Fish Community Structures in French Rivers*

Description

Results from studies examining changes in the abundance of fish species in French rivers.

Usage

```
dat.maire2019
```

Format

The object is a list containing a data frame called `dat` that contains the following columns and distance matrix called `dmat`:

site	`character`	study site
station	`character`	sampling station at site
site_station	`character`	site and station combined
s1	`numeric`	Mann-Kendal trend statistic for relative abundance of non-local species
vars1	`numeric`	corresponding sampling variance (corrected for temporal autocorrelation)
s2	`numeric`	Mann-Kendal trend statistic for relative abundance of northern species
vars2	`numeric`	corresponding sampling variance (corrected for temporal autocorrelation)
s3	`numeric`	Mann-Kendal trend statistic for relative abundance of non-native species
vars3	`numeric`	corresponding sampling variance (corrected for temporal autocorrelation)
const	`numeric`	constant value of 1

Details

The dataset includes the results from 35 sampling stations (at 11 sites along various French rivers) examining the abundance of various fish species over time (i.e., over 19-37 years, all until 2015). The temporal trend in these abundance data was quantified in terms of Mann-Kendal trend statistics, with positive values indicating monotonically increasing trends. The corresponding sampling variances were corrected for the temporal autocorrelation in the data (Hamed & Rao, 1998).

The distance matrix `dmat` indicates the distance of the sampling stations (1-423 river-km). For stations not connected through the river network, a high distance value of 10,000 river-km was set (effectively forcing the spatial correlation to be 0 for such stations).

The dataset can be used to illustrate a meta-analysis allowing for spatial correlation in the outcomes.

Source

Maire, A., Thierry, E., Viechtbauer, W., & Daufresne, M. (2019). Poleward shift in large-river fish communities detected with a novel meta-analysis framework. *Freshwater Biology*, **64**(6), 1143–1156. https://doi.org/10.1111/fwb.13291

References

Hamed, K. H., & Rao, A. R. (1998). A modified Mann-Kendall trend test for autocorrelated data. *Journal of Hydrology*, **204**(1-4), 182–196. https://doi.org/10.1016/S0022-1694(97)00125-X

Examples

```
### copy data into 'dat' and examine data
dat <- dat.maire2019$dat
dat

### copy distance matrix into 'dmat' and examine first 5 rows/columns
dmat <- dat.maire2019$dmat
dmat[1:5,1:5]

### fit a standard random-effects model ignoring spatial correlation
res1 <- rma.mv(s1, vars1, random = ~ 1 | site_station, data=dat)
res1

### fit model allowing for spatial correlation
res2 <- rma.mv(s1, vars1, random = ~ site_station | const, struct="SPGAU",
               data=dat, dist=list(dmat), control=list(rho.init=10))
res2

### add random effects for sites and stations within sites
res3 <- rma.mv(s1, vars1, random = list(~ 1 | site/station, ~ site_station | const), struct="SPGAU",
               data=dat, dist=list(dmat), control=list(rho.init=10))
res3

### likelihood ratio tests comparing the models
anova(res1, res2)
anova(res2, res3)

### profile likelihood plots for model res2
profile(res2, cline=TRUE)

### effective range (river-km for which the spatial correlation is >= .05)
sqrt(3) * res2$rho
```

---

dat.mcdaniel1994 *Studies on the Validity of Employment Interviews*

Description

Results from 160 studies on the correlation between employment interview assessments and job performance.

Usage

```
dat.mcdaniel1994
```

Format

The data frame contains the following columns:

study	`numeric`	study number
ni	`numeric`	sample size of the study
ri	`numeric`	observed correlation
type	`character`	interview type (j = job-related, s = situational, p = psychological)
struct	`character`	interview structure (u = unstructured, s = structured)

Details

The 160 studies provide data in terms of the correlation between employment interview performance and actual job performance. In addition, the interview type and the interview structure are indicated.

McDaniel et al. (1994) describe the interview type and structure variables as follows. "Questions in situational interviews [...] focus on the individual's ability to project what his or her behavior would be in a given situation. [...] Job-related interviews are those in which the interviewer is a personnel officer or hiring authority and the questions attempt to assess past behaviors and job-related information, but most questions are not considered situational. Psychological interviews are conducted by a psychologist, and the questions are intended to assess personal traits, such as dependability." In structured interviews, "the questions and acceptable responses were specified in advance and the responses were rated for appropriateness of content. [...] Unstructured interviews gather applicant information in a less systematic manner than do structured interviews. Although the questions may be specified in advance, they usually are not, and there is seldom a formalized scoring guide. Also, all persons being interviewed are not typically asked the same questions."

The goal of the meta-analysis was to examine the overall criterion-related validity of employment interviews and to examine whether the validity depends on the type and structure of the interview.

The data in this dataset were obtained from Table A.2 in Rothstein, Sutton, and Borenstein (2005, p. 325-329). Note that the `type` and `struct` variables contain some `NA`s.

Source

Rothstein, H. R., Sutton, A. J., & Borenstein, M. (Eds.). (2005). *Publication bias in meta-analysis: Prevention, assessment, and adjustments*. Chichester, England: Wiley.

References

McDaniel, M. A., Whetzel, D. L., Schmidt, F. L., & Maurer, S. D. (1994). The validity of employment interviews: A comprehensive review and meta-analysis. *Journal of Applied Psychology*, **79**(4), 599–616. https://doi.org/10.1037/0021-9010.79.4.599

Examples

```
### copy data into 'dat'
dat <- dat.mcdaniel1994

### calculate r-to-z transformed correlations and corresponding sampling variances
dat <- escalc(measure="ZCOR", ri=ri, ni=ni, data=dat)
dat

### meta-analysis of the transformed correlations using a random-effects model
res <- rma(yi, vi, data=dat)
res

### average correlation with 95% CI
predict(res, transf=transf.ztor)

### mixed-effects model with interview type as factor
### note: job-related interviews is the reference level
rma(yi, vi, mods = ~ factor(type), data=dat)

### mixed-effects model with interview structure as factor
### note: structured interviews is the reference level
rma(yi, vi, mods = ~ factor(struct), data=dat)

### note: the interpretation of the results is difficult since all
### situational interviews were structured, almost all psychological
### interviews were unstructured, and actually for the majority of
### the psychological interviews it was unknown whether the interview
### was structured or unstructured
table(dat$type, dat$struct, useNA="always")

### meta-analysis of raw correlations using a random-effects model
res <- rma(measure="COR", ri=ri, ni=ni, data=dat.mcdaniel1994)
res
```

`dat.molloy2014` *Studies on the Relationship between Conscientiousness and Medication Adherence*

Description

Results from 16 studies on the correlation between conscientiousness and medication adherence.

Usage

```
dat.molloy2014
```

Format

The data frame contains the following columns:

authors	`character`	study authors
year	`numeric`	publication year
ni	`numeric`	sample size of the study
ri	`numeric`	observed correlation
controls	`character`	number of variables controlled for
design	`character`	whether a cross-sectional or prospective design was used
a_measure	`character`	type of adherence measure (self-report or other)
c_measure	`character`	type of conscientiousness measure (NEO or other)
meanage	`numeric`	mean age of the sample
quality	`numeric`	methodological quality

Details

Conscientiousness, one of the big-5 personality traits, can be defined as "socially prescribed impulse control that facilitates task- and goal-directed behaviour, such as thinking before acting, delaying gratification, following norms and rules and planning, organising and prioritising tasks" (John & Srivastava, 1999). Conscientiousness has been shown to be related to a number of health-related behaviors (e.g., tobacco/alcohol/drug use, diet and activity patterns, risky behaviors). A recent meta-analysis by Molloy et al. (2014) examined to what extent conscientiousness is related to medication adherence, that is, the extent to which (typically chronically ill) patients follow a prescribed medication regimen (e.g., taking a daily dose of a cholesterol lowering drug in patients with high LDL serum cholesterol levels). The results from the 16 studies included in this meta-analysis are provided in this dataset.

Variable `a_measure` indicates whether adherence was measured based on self-reports or a more ‘objective’ measure (e.g., electronic monitoring of pill bottle openings, pill counts). Variable c_measure indicates whether conscientiousness was measured with some version of the NEO personality inventory or some other scale. Methodological quality was scored by the authors on a 1 to 4 scale with higher scores indicating higher quality (see article for details on how this score was derived).

Source

Molloy, G. J., O’Carroll, R. E., & Ferguson, E. (2014). Conscientiousness and medication adherence: A meta-analysis. *Annals of Behavioral Medicine*, **47**(1), 92–101. https://doi.org/10.1007/s12160-013-9524-4

References

John, O. P., & Srivastava, S. (1999). The Big Five Trait taxonomy: History, measurement, and theoretical perspectives. In L. A. Pervin & O. P. John (Eds.), *Handbook of personality: Theory and research* (2nd ed., pp. 102-138). New York: Guilford Press.

Examples

```
### copy data into 'dat'
dat <- dat.molloy2014

### calculate r-to-z transformed correlations and corresponding sampling variances
dat <- escalc(measure="ZCOR", ri=ri, ni=ni, data=dat, slab=paste(authors, year, sep=", "))
dat

### meta-analysis of the transformed correlations using a random-effects model
res <- rma(yi, vi, data=dat)
res

### average correlation with 95% CI
predict(res, digits=3, transf=transf.ztor)

### forest plot
forest(res, addpred=TRUE, xlim=c(-1.6,1.6), atransf=transf.ztor,
       at=transf.rtoz(c(-.4,-.2,0,.2,.4,.6)), digits=c(2,1), cex=.8,
       header="Author(s), Year")

### funnel plot
funnel(res)
```

dat.moura2021 *Studies on Assortative Mating*

Description

Results from 457 studies on assortative mating in various species.

Usage

```
dat.moura2021
```

Format

The object is a list containing a data frame called `dat` that contains the following columns and a phylogenetic tree called `tree`:

study.id	`character`	study id
effect.size.id	`numeric`	effect size id
species	`character`	species
species.id	`character`	species id (as in the Open Tree of Life reference taxonomy)
subphylum	`character`	the subphyla of the species
phylum	`character`	the phyla of the species
assortment.trait	`character`	the measure of body size
trait.dimensions	`character`	dimensionality of the measure
field.collection	`character`	whether data were collected in the field
publication.year	`numeric`	publication year of the study
pooled.data	`character`	whether data were pooled either spatially and/or temporally
spatially.pooled.data	`character`	whether data were pooled spatially
temporally.pooled.data	`character`	whether data were pooled temporally
ri	`numeric`	correlation coefficient
ni	`numeric`	sample size

Blah.

Details

The 457 studies included in this dataset provide 1828 correlation coefficients describing the similarity in some measure of body size in mating couples in 341 different species.

Source

Rios Moura, R., Oliveira Gonzaga, M., Silva Pinto, N., Vasconcellos-Neto, J., & Requena, G. S. (2021). Assortative mating in space and time: Patterns and biases. *Ecology Letters*, **24**(5), 1089–1102. https://doi.org/10.1111/ele.13690

References

Hadfield, J. D., & Nakagawa, S. (2010). General quantitative genetic methods for comparative biology: Phylogenies, taxonomies and multi-trait models for continuous and categorical characters. *Journal of Evolutionary Biology*, **23**(3), 494–508. https://doi.org/10.1111/j.1420-9101.2009.01915.x

Nakagawa, S., & Santos, E. S. A. (2012). Methodological issues and advances in biological meta-analysis. *Evolutionary Ecology*, **26**(5), 1253–1274. https://doi.org/10.1007/s10682-012-9555-5

Examples

```
### copy data into 'dat' and examine data
dat <- dat.moura2021$dat
```

```
head(dat)

### calculate r-to-z transformed correlations and corresponding sampling variances
dat <- escalc(measure="ZCOR", ri=ri, ni=ni, data=dat)

## Not run:
### load 'ape' package
require(ape)

### copy tree to 'tree'
tree <- dat.moura2021$tree

### turn tree into an ultrametric one
tree <- compute.brlen(tree)

### compute phylogenetic correlation matrix
A <- vcv(tree, corr=TRUE)

### make copy of the species.id variable
dat$species.id.phy <- dat$species.id

### fit multilevel phylogenetic meta-analytic model
res <- rma.mv(yi, vi,
  random = list(~ 1 | study.id, ~ 1 | effect.size.id, ~ 1 | species.id, ~ 1 | species.id.phy),
   R=list(species.id.phy=A), data=dat)
res

### examine if spatial and/or temporal pooling of data tends to yield larger correlations
res <- rma.mv(yi, vi,
   mods = ~ spatially.pooled.data * temporally.pooled.data,
  random = list(~ 1 | study.id, ~ 1 | effect.size.id, ~ 1 | species.id, ~ 1 | species.id.phy),
   R=list(species.id.phy=A), data=dat)
res

### estimated average correlation without pooling, when pooling spatially,
### when pooling temporally, and when pooling spatially and temporally
predict(res, newmods = rbind(c(0,0,0),c(1,0,0),c(0,1,0),c(1,1,1)), transf=transf.ztor, digits=2)

## End(Not run)
```

`dat.nielweise2007` *Studies on Anti-Infective-Treated Central Venous Catheters for Prevention of Catheter-Related Bloodstream Infections*

Description

Results from 18 studies comparing the risk of catheter-related bloodstream infection when using anti-infective-treated versus standard catheters in the acute care setting.

Usage

```
dat.nielweise2007
```

Format

The data frame contains the following columns:

study	`numeric`	study number
author	`character`	(first) author
year	`numeric`	publication year
ai	`numeric`	number of CRBSIs in patients receiving an anti-infective catheter
n1i	`numeric`	number of patients receiving an anti-infective catheter
ci	`numeric`	number of CRBSIs in patients receiving a standard catheter
n2i	`numeric`	number of patients receiving a standard catheter

Details

The use of a central venous catheter may lead to a catheter-related bloodstream infection (CRBSI), which in turn increases the risk of morbidity and mortality. Anti-infective-treated catheters have been developed that are meant to reduce the risk of CRBSIs. Niel-Weise et al. (2007) conducted a meta-analysis of studies comparing infection risk when using anti-infective-treated versus standard catheters in the acute care setting. The results from 18 such studies are included in this dataset.

The dataset was used in the article by Stijnen et al. (2010) to illustrate various generalized linear mixed-effects models for the meta-analysis of proportions and odds ratios (see 'References').

Source

Niel-Weise, B. S., Stijnen, T., & van den Broek, P. J. (2007). Anti-infective-treated central venous catheters: A systematic review of randomized controlled trials. *Intensive Care Medicine*, **33**(12), 2058–2068. https://doi.org/10.1007/s00134-007-0897-3

References

Stijnen, T., Hamza, T. H., & Ozdemir, P. (2010). Random effects meta-analysis of event outcome in the framework of the generalized linear mixed model with applications in sparse data. *Statistics in Medicine*, **29**(29), 3046–3067. https://doi.org/10.1002/sim.4040

Examples

```
### copy data into 'dat' and examine data
dat <- dat.nielweise2007
dat

### standard (inverse-variance) random-effects model
res <- rma(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, drop00=TRUE)
print(res, digits=3)
predict(res, transf=exp, digits=2)

### random-effects conditional logistic model
## Not run:
```

```
res <- rma.glmm(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, model="CM.EL")
print(res, digits=3)
predict(res, transf=exp, digits=2)
## End(Not run)
```

---

dat.nielweise2008 *Studies on Anti-Infective-Treated Central Venous Catheters for Prevention of Catheter-Related Bloodstream Infections*

Description

Results from 18 studies comparing the risk of catheter-related bloodstream infection when using anti-infective-treated versus standard catheters for total parenteral nutrition or chemotherapy.

Usage

```
dat.nielweise2008
```

Format

The data frame contains the following columns:

study	`numeric`	study number
authors	`character`	study authors
year	`numeric`	publication year
x1i	`numeric`	number of CRBSIs in patients receiving an anti-infective catheter
t1i	`numeric`	total number of catheter days for patients receiving an anti-infective catheter
x2i	`numeric`	number of CRBSIs in patients receiving a standard catheter
t2i	`numeric`	total number of catheter days for patients receiving a standard catheter

Details

The use of a central venous catheter may lead to a catheter-related bloodstream infection (CRBSI), which in turn increases the risk of morbidity and mortality. Anti-infective-treated catheters have been developed that are meant to reduce the risk of CRBSIs. Niel-Weise et al. (2008) conducted a meta-analysis of studies comparing infection risk when using anti-infective-treated versus standard catheters for total parenteral nutrition or chemotherapy. The results from 9 such studies are included in this dataset.

The dataset was used in the article by Stijnen et al. (2010) to illustrate various generalized linear mixed-effects models for the meta-analysis of incidence rates and incidence rate ratios (see ‘References’).

Source

Niel-Weise, B. S., Stijnen, T., & van den Broek, P. J. (2008). Anti-infective-treated central venous catheters for total parenteral nutrition or chemotherapy: A systematic review. *Journal of Hospital Infection*, **69**(2), 114–123. https://doi.org/10.1016/j.jhin.2008.02.020

References

Stijnen, T., Hamza, T. H., & Ozdemir, P. (2010). Random effects meta-analysis of event outcome in the framework of the generalized linear mixed model with applications in sparse data. *Statistics in Medicine*, **29**(29), 3046–3067. https://doi.org/10.1002/sim.4040

Examples

```
### copy data into 'dat' and examine data
dat <- dat.nielweise2008
dat

### standard (inverse-variance) random-effects model
res <- rma(measure="IRR", x1i=x1i, t1i=t1i, x2i=x2i, t2i=t2i, data=dat)
print(res, digits=3)
predict(res, transf=exp, digits=2)

### random-effects conditional Poisson model
## Not run:
res <- rma.glmm(measure="IRR", x1i=x1i, t1i=t1i, x2i=x2i, t2i=t2i, data=dat, model="CM.EL")
print(res, digits=3)
predict(res, transf=exp, digits=2)
## End(Not run)
```

`dat.normand1999` *Studies on the Length of Hospital Stay of Stroke Patients*

Description

Results from 9 studies on the length of the hospital stay of stroke patients under specialized care and under conventional/routine (non-specialist) care.

Usage

```
dat.normand1999
```

Format

The data frame contains the following columns:

study	`numeric`	study number
source	`character`	source of data
n1i	`numeric`	number of patients under specialized care
m1i	`numeric`	mean length of stay (in days) under specialized care
sd1i	`numeric`	standard deviation of the length of stay under specialized care
n2i	`numeric`	number of patients under routine care
m2i	`numeric`	mean length of stay (in days) under routine care
sd2i	`numeric`	standard deviation of the length of stay under routine care

Details

The 9 studies provide data in terms of the mean length of the hospital stay (in days) of stroke patients under specialized care and under conventional/routine (non-specialist) care. The goal of the meta-analysis was to examine the hypothesis whether specialist stroke unit care will result in a shorter length of hospitalization compared to routine management.

Source

Normand, S. T. (1999). Meta-analysis: Formulating, evaluating, combining, and reporting. *Statistics in Medicine*, **18**(3), 321–359. https://doi.org/10.1002/(sici)1097-0258(19990215)18:3<321::aid-sim28>3.0.co;2-p

Examples

```
### copy data into 'dat'
dat <- dat.normand1999

### calculate mean differences and corresponding sampling variances
dat <- escalc(measure="MD", m1i=m1i, sd1i=sd1i, n1i=n1i, m2i=m2i, sd2i=sd2i, n2i=n2i, data=dat)
dat

### meta-analysis of mean differences using a random-effects model
res <- rma(yi, vi, data=dat)
res

### meta-analysis of standardized mean differences using a random-effects model
res <- rma(measure="SMD", m1i=m1i, sd1i=sd1i, n1i=n1i, m2i=m2i, sd2i=sd2i, n2i=n2i,
           data=dat, slab=source)
res

### draw forest plot
forest(res, xlim=c(-7,5), alim=c(-3,1), cex=.8, header="Study/Source")
```

`dat.obrien2003` *Studies on the Relationship Between BMI and Risk of Preeclampsia*

Description

Results from 13 studies on the relationship between maternal body mass index (BMI) and the risk of preeclampsia.

Usage

```
dat.obrien2003
```

Format

The data frame contains the following columns:

study	`numeric`	study id
author	`character`	(first) author of the study
year	`numeric`	publication year
ref	`numeric`	reference number
ch	`character`	exclusion due to chronic hypertension (yes/no)
dm	`character`	exclusion due to diabetes mellitus (yes/no)
mg	`character`	exclusion due to multiple gestation (yes/no)
bmi.lb	`numeric`	lower bound of the BMI interval
bmi.ub	`numeric`	upper bound of the BMI interval
bmi	`numeric`	midpoint of the BMI interval
cases	`numeric`	number of preeclampsia cases in the BMI group
total	`numeric`	number of individuals in the BMI group

Details

The dataset includes the results from 13 studies examining the relationship between maternal body mass index (BMI) and the risk of preeclampsia. For each study, results are given in terms of the number of preeclampsia cases within two or more groups defined by the lower and upper BMI bounds as shown in the dataset (`NA` means that the interval is either open to the left or right). The `bmi` variable is the interval midpoint as defined by O'Brien et al. (2003).

Source

O'Brien, T. E., Ray, J. G., & Chan, W.-S. (2003). Maternal body mass index and the risk of preeclampsia: A systematic overview. *Epidemiology*, **14**(3), 368–374. https://doi.org/10.1097/00001648-200305000-00020

Examples

```
### copy data into 'dat'
dat <- dat.obrien2003
dat

### restructure the data into a wide format
dat2 <- to.wide(dat, study="study", grp="grp", ref=1, grpvars=c("bmi","cases","total"),
                addid=FALSE, adddesign=FALSE, postfix=c(1,2))
dat2

### calculate log risk ratios and corresponding sampling variances
dat2 <- escalc(measure="RR", ai=cases1, n1i=total1, ci=cases2, n2i=total2, data=dat2)
dat2

### forest plot of the risk ratios
dd <- c(0,diff(dat2$study))
dd[dd > 0] <- 1
rows <- (1:nrow(dat2)) + cumsum(dd)
rows <- 1 + max(rows) - rows
slabs <- mapply(function(x,y,z) as.expression(bquote(.(x)^.(y)~.(z))),
                dat2$author, dat2$ref, dat2$year)
with(dat2, forest(yi, vi, header=TRUE, slab=slabs, xlim=c(-7,5.5), fonts="mono", cex=0.8,
   psize=1, pch=19, efac=0, rows=rows, ylim=c(0,max(rows)+3), yaxs="i",
  atransf=exp, at=log(c(.05,0.1,0.2,0.5,1,2,5,10,20)), ilab=comp, ilab.xpos=-4, ilab.pos=4))
text(-4.4, max(rows)+2, "Comparison", font=2, cex=0.8, pos=4)

### within-study mean center the BMI variable
dat$bmicent <- dat$bmi - ave(dat$bmi, dat$study)

### compute the proportion of preeclampsia cases and corresponding sampling variances
dat <- escalc(measure="PR", xi=cases, ni=total, data=dat)

### convert the proportions to percentages (and convert the variances accordingly)
dat$yi <- dat$yi*100
dat$vi <- dat$vi*100^2

### fit multilevel meta-regression model to examine the relationship between the
### (centered) BMI variable and the risk of preeclampsia
res <- rma.mv(yi, vi, mods = ~ bmicent, random = ~ 1 | study/grp, data=dat)
res

### draw scatterplot with regression line
res$slab <- dat$ref
regplot(res, xlab=expression("Within-Study Mean Centered BMI"~(kg/m^2)),
        ylab="Preeclampsia Prevalence (%)", las=1, bty="l",
        at=seq(0,18,by=2), olim=c(0,100), psize=2, bg="gray90",
        label=TRUE, offset=0, labsize=0.6)
```

dat.pagliaro1992 *Studies on the Effectiveness of Nonsurgical Treatments in Cirrhosis*

Description

Results from 26 trials examining the effectiveness of beta-blockers and sclerotherapy for the prevention of first bleeding in patients with cirrhosis

Usage

```
dat.pagliaro1992
```

Format

The data frame contains the following columns:

study	`numeric`	study id
trt	`character`	either beta-blockers, sclerotherapy, or control
xi	`numeric`	number of patients with first bleeding
ni	`numeric`	number of patients treated

Details

The dataset includes the results from 26 randomized controlled trials examining the effectiveness of nonsurgical treatments for the prevention of first bleeding in patients with cirrhosis. Patients were either treated with beta-blockers, endoscopic sclerotherapy, or with a nonactive treatment (control). Two trials included all three treatment conditions, 7 trials compared beta-blockers against control, and 17 trials compared sclerotherapy against control. The dataset has been used in various papers to illustrate methods for conducting a network meta-analysis / mixed treatment comparison.

Source

Pagliaro, L., D'Amico, G., Sörensen, T. I. A., Lebrec, D., Burroughs, A. K., Morabito, A., Tiné, F., Politi, F., & Traina, M. (1992). Prevention of first bleeding in cirrhosis: A meta-analysis of randomized trials of nonsurgical treatment. *Annals of Internal Medicine*, **117**(1), 59–70. https://doi.org/10.7326/0003-4819-117-1-59

Examples

```
### copy data into 'dat' and examine data
dat <- dat.pagliaro1992
dat

### restructure dataset to a contrast-based format
dat.c <- to.wide(dat, study="study", grp="trt", grpvars=3:4)
dat.c

### Mantel-Haenszel results for beta-blockers and sclerotherapy versus control, respectively
rma.mh(measure="OR", ai=xi.1, n1i=ni.1, ci=xi.2, n2i=ni.2,
       data=dat.c, subset=(trt.1=="beta-blockers"), digits=2)
rma.mh(measure="OR", ai=xi.1, n1i=ni.1, ci=xi.2, n2i=ni.2,
       data=dat.c, subset=(trt.1=="sclerotherapy"), digits=2)

### calculate log odds for each study arm
dat <- escalc(measure="PLO", xi=xi, ni=ni, data=dat)
dat

### turn treatment variable into factor and set reference level
dat$trt <- relevel(factor(dat$trt), ref="control")

### add a space before each level (this makes the output a bit more legible)
levels(dat$trt) <- paste0(" ", levels(dat$trt))

### network meta-analysis using an arm-based random-effects model with fixed study effects
### (by setting rho=1/2, tau^2 reflects the amount of heterogeneity for all treatment comparisons)
res <- rma.mv(yi, vi, mods = ~ factor(study) + trt - 1, random = ~ trt | study, rho=1/2, data=dat)
res

### average odds ratio comparing beta-blockers and sclerotherapy versus control, respectively
predict(res, newmods=c(rep(0,26), 1, 0), transf=exp, digits=2)
predict(res, newmods=c(rep(0,26), 0, 1), transf=exp, digits=2)

### average odds ratio comparing beta-blockers versus sclerotherapy
```

```
predict(res, newmods=c(rep(0,26), 1, -1), transf=exp, digits=2)
```

---

dat.pignon2000 *Studies on the Effectiveness of Locoregional Treatment plus Chemotherapy for Head and Neck Squamous-Cell Carcinoma*

Description

Results from studies examining mortality risk in patients with nonmetastatic head and neck squamous-cell carcinoma receiving either locoregional treatment plus chemotherapy versus locoregional treatment alone.

Usage

```
dat.pignon2000
```

Format

The data frame contains the following columns:

id	`numeric`	study id number
trial	`character`	trial abbreviation
OmE	`numeric`	observed minus expected number of deaths in the locoregional treatment plus chemotherapy group
V	`numeric`	corresponding variance
grp	`numeric`	timing of chemotherapy: 1 = adjuvant, 2 = neoadjuvant, 3 = concomitant

Details

The purpose of this meta-analysis was to examine the mortality risk in patients with nonmetastatic head and neck squamous-cell carcinoma receiving either locoregional treatment plus chemotherapy versus locoregional treatment alone. For 65 trials, the dataset provides the observed minus expected number of deaths and corresponding variances in the locoregional treatment plus chemotherapy group. Based on these values, we can estimate the log hazard ratios with `OmE/V` and the corresponding sampling variance with `1/V`.

The trials were also divided according to the timing of the chomotherapy: (1) adjuvant, after the locoregional treatment, (2) neoadjuvant, before the locoregional treatment, and (3) concomitant, chemotherapy given concomitantly or alternating with radiotherapy.

Source

Pignon, J. P., Bourhis, J., Domenge, C., & Designe, L. (2000). Chemotherapy added to locoregional treatment for head and neck squamous-cell carcinoma: Three meta-analyses of updated individual data. *Lancet*, **355**(9208), 949–955. https://doi.org/10.1016/S0140-6736(00)90011-4

Examples

```
### copy data into 'dat'
```

```
dat <- dat.pignon2000

### calculate log hazard ratios and sampling variances
dat$yi <- with(dat, OmE/V)
dat$vi <- with(dat, 1/V)
dat

### meta-analysis based on all 65 trials
res <- rma(yi, vi, data=dat, method="FE", digits=2)
res
predict(res, transf=exp)

### only adjuvant trials
res <- rma(yi, vi, data=dat, method="FE", subset=grp==1, digits=2)
res
predict(res, transf=exp)

### only neoadjuvant trials
res <- rma(yi, vi, data=dat, method="FE", subset=grp==2, digits=2)
res
predict(res, transf=exp)

### only concomitant trials
res <- rma(yi, vi, data=dat, method="FE", subset=grp==3, digits=2)
res
predict(res, transf=exp)
```

dat.pritz1997 *Studies on the Effectiveness of Hyperdynamic Therapy for Treating Cerebral Vasospasm*

Description

Results from 14 studies on the effectiveness of hyperdynamic therapy for treating cerebral vasospasm.

Usage

```
dat.pritz1997
```

Format

The data frame contains the following columns:

study	`numeric`	study number
authors	`character`	study authors
xi	`numeric`	number of patients that improved with hyperdynamic therapy
ni	`numeric`	total number of patients treated

Details

As described in Zhou et al. (1999), "hyperdynamic therapy refers to induced hypertension and hypervolaemia (volume expansion) to treat ischaemic symptoms due to vasospasm, and the success of this therapy is defined as clinical improvement in terms of neurologic deficits." For each study that was included in the meta-analysis, the dataset includes information on the number of patients that improved under this form of therapy and the total number of patients that were treated. The goal of the meta-analysis is to estimate the true (average) success rate of hyperdynamic therapy.

Source

Zhou, X.-H., Brizendine, E. J., & Pritz, M. B. (1999). Methods for combining rates from several studies. *Statistics in Medicine*, **18**(5), 557–566. https://doi.org/10.1002/(SICI)1097-0258(19990315)18:5<557::AID-SIM53>3.0.CO;2-F

References

Pritz, M. B. (1997). Treatment of cerebral vasospasm due to aneurysmal subarachnoid hemorrhage: Past, present, and future of hyperdynamic therapy. *Neurosurgery Quarterly*, **7**(4), 273–285.

Examples

```
### copy data into 'dat' and examine data
dat <- dat.pritz1997
dat

### computation of "weighted average" in Zhou et al. (1999), Table IV
dat <- escalc(measure="PR", xi=xi, ni=ni, data=dat, add=0)
theta.hat    <- sum(dat$ni * dat$yi) / sum(dat$ni)
se.theta.hat <- sqrt(sum(dat$ni^2 * dat$vi) / sum(dat$ni)^2)
ci.lb        <- theta.hat - 1.96 * se.theta.hat
ci.ub        <- theta.hat + 1.96 * se.theta.hat
round(c(estimate = theta.hat, se = se.theta.hat, ci.lb = ci.lb, ci.ub = ci.ub), 4)

### this is identical to a FE model with sample size weights
rma(yi, vi, weights=ni, method="FE", data=dat)

### random-effects model with raw proportions
dat <- escalc(measure="PR", xi=xi, ni=ni, data=dat)
res <- rma(yi, vi, data=dat)
predict(res)

### random-effects model with logit transformed proportions
dat <- escalc(measure="PLO", xi=xi, ni=ni, data=dat)
res <- rma(yi, vi, data=dat)
predict(res, transf=transf.ilogit)

### mixed-effects logistic regression model
res <- rma.glmm(measure="PLO", xi=xi, ni=ni, data=dat)
predict(res, transf=transf.ilogit)
```

dat.raudenbush1985 *Studies on Assessing the Effects of Teacher Expectations on Pupil IQ*

Description

Results from 19 studies examining how teachers' expectations about their pupils can influence actual IQ levels.

Usage

```
dat.raudenbush1985
```

Format

The data frame contains the following columns:

study	`numeric`	study number
author	`character`	study author(s)
year	`numeric`	publication year
weeks	`numeric`	weeks of contact prior to expectancy induction
setting	`character`	whether tests were group or individually administered
tester	`character`	whether test administrator was aware or blind
n1i	`numeric`	sample size of experimental group
n2i	`numeric`	sample size of control group
yi	`numeric`	standardized mean difference
vi	`numeric`	corresponding sampling variance

Details

In the so-called 'Pygmalion study' (Rosenthal & Jacobson, 1968), "all of the predominantly poor children in the so-called Oak elementary school were administered a test pretentiously labeled the 'Harvard Test of Inflected Acquisition.' After explaining that this newly designed instrument had identified those children most likely to show dramatic intellectual growth during the coming year, the experimenters gave the names of these 'bloomers' to the teachers. In truth, the test was a traditional IQ test and the 'bloomers' were a randomly selected 20% of the student population. After retesting the children 8 months later, the experimenters reported that those predicted to bloom had in fact gained significantly more in total IQ (nearly 4 points) and reasoning IQ (7 points) than the control group children. Further, at the end of the study, the teachers rated the experimental children as intellectually more curious, happier, better adjusted, and less in need of approval than their control group peers" (Raudenbush, 1984).

In the following years, a series of studies were conducted attempting to replicate this rather controversial finding. However, the great majority of those studies were unable to demonstrate a statistically significant difference between the two experimental groups in terms of IQ scores. Raudenbush (1984) conducted a meta-analysis based on 19 such studies to further examine the evidence for the existence of the 'Pygmalion effect'. The dataset includes the results from these studies.

The outcome measure used for the meta-analysis was the standardized mean difference (`yi`), with positive values indicating that the supposed 'bloomers' had, on average, higher IQ scores than those in the control group. The weeks variable indicates the number of weeks of prior contact between teachers and students before the expectancy induction. Testing was done either in a group setting or individually, which is indicated by the setting variable. Finally, the tester variable indicates whether the test administrators were either aware or blind to the researcher-provided designations of the children's intellectual potential.

The data in this dataset were obtained from Raudenbush and Bryk (1985) with information on the setting and tester variables extracted from Raudenbush (1984).

Source

Raudenbush, S. W. (1984). Magnitude of teacher expectancy effects on pupil IQ as a function of the credibility of expectancy induction: A synthesis of findings from 18 experiments. *Journal of Educational Psychology*, **76**(1), 85–97. https://doi.org/10.1037/0022-0663.76.1.85

Raudenbush, S. W., & Bryk, A. S. (1985). Empirical Bayes meta-analysis. *Journal of Educational Statistics*, **10**(2), 75–98. https://doi.org/10.3102/10769986010002075

Examples

```
### copy data into 'dat' and examine data
dat <- dat.raudenbush1985
dat

### random-effects model
res <- rma(yi, vi, data=dat)
res

### create weeks variable where values larger than 3 are set to 3
dat$weeks.c <- ifelse(dat$week > 3, 3, dat$week)

### mixed-effects model with weeks.c variable as moderator
res <- rma(yi, vi, mods=~weeks.c, data=dat, digits=3)
res
```

---

dat.riley2003 *Studies on MYC-N as a Prognostic Marker for Neuroblastoma*

Description

Results from 81 studies examining overall and disease-free survival in neuroblastoma patients with amplified versus normal MYC-N protein levels.

Usage

```
dat.riley2003
```

Format

The data frame contains the following columns:

study	`numeric`	study number
yi	`numeric`	log hazard ratio of the outcome in those with amplified versus normal MYC-N protein levels
sei	`numeric`	standard error of the log hazard ratio
outcome	`character`	outcome (OS = overall survival; DFS = disease-free survival)

Details

The meta-analysis by Riley et al. (2003) examined a variety of prognostic markers for overall and disease-free survival in patients with neuroblastoma. One of the markers examined was amplified levels of the MYC-N protein, with is associated with poorer outcomes.

The dataset given here was extracted from Riley (2011) and has been used in several other publications (e.g., Riley et al., 2004, 2007). The dataset provides the (log) hazard ratios (and corresponding standard errors) with respect to these two outcomes in 81 studies, with positive values indicating a greater risk of death (for OS) or disease recurrence/death (for DFS) for patients with high MYC-N levels compared to those with normal/low levels. Note that information on both outcomes could only be extracted from 17 studies (39 studies only provided sufficient information to extract the OS estimate, while 25 studies only allowed for extraction of the DFS estimate).

Source

Riley, R. D., Sutton, A. J., Abrams, K. R., & Lambert, P. C. (2004). Sensitivity analyses allowed more appropriate and reliable meta-analysis conclusions for multiple outcomes when missing data was present. *Journal of Clinical Epidemiology*, **57**(9), 911–924. https://doi.org/10.1016/j.jclinepi.2004.01.018

Riley, R. D., Abrams, K. R., Lambert, P. C., Sutton, A. J., & Thompson, J. R. (2007). An evaluation of bivariate random-effects meta-analysis for the joint synthesis of two correlated outcomes. *Statistics in Medicine*, **26**(1), 78–97. https://doi.org/10.1002/sim.2524

Riley, R. D. (2011). Erratum: An evaluation of bivariate random-effects meta-analysis for the joint synthesis of two correlated outcomes. *Statistics in Medicine*, **30**(4), 400. https://doi.org/10.1002/sim.4100

References

Riley, R. D., Burchill, S. A., Abrams, K. R., Heney, D., Lambert, P. C., Jones, D. R., Sutton, A. J., Young, B., Wailoo, A. J., & Lewis, I. J. (2003). A systematic review and evaluation of the use of tumour markers in paediatric oncology: Ewing's sarcoma and neuroblastoma. *Health Technology Assessment*, **7**(5), 1–162. https://doi.org/10.3310/hta7050

Examples

```
### copy data into 'dat' and examine data
dat <- dat.riley2003
dat

### random-effects model analysis for outcome DFS
res <- rma(yi, sei=sei, data=dat, subset=(outcome == "DFS"), method="DL")
res
predict(res, transf=exp, digits=2)

### random-effects model analysis for outcome OS
res <- rma(yi, sei=sei, data=dat, subset=(outcome == "OS"), method="DL")
```

```
res
predict(res, transf=exp, digits=2)
```

dat.senn2013 *Studies on the Effectiveness of Glucose-Lowering Agents*

Description

Results from 26 trials examining the effectiveness of glucose-lowering agents in patients with type 2 diabetes

Usage

```
dat.senn2013
```

Format

The data frame contains the following columns:

study	`character`	(first) author and year of study
ni	`numeric`	sample size of the study arm
treatment	`character`	treatment given
comment	`character`	whether figures given are based on raw values at outcome or on change from baseline
mi	`numeric`	mean score
sdi	`numeric`	standard deviation

Details

The dataset includes the results from 26 randomized controlled trials examining the effectiveness of adding various oral glucose-lowering agents to a baseline sulfonylurea therapy in patients with type 2 diabetes. The outcome measured in the studies was either the mean HbA1c level at follow-up or the mean change in HbA1c level from baseline to follow-up. A total of 10 different treatment types were examined in these studies: acarbose, benfluorex, metformin, miglitol, pioglitazone, placebo, rosiglitazone, sitagliptin, sulfonylurea alone, and vildagliptin. One study included three treatment arms (Willms, 1999), while the rest of the studies included two treatment arms (hence, the dataset includes the results from 53 treatment arms).

The data can be used for a network meta-analysis, either using an arm-based or a contrast-based model. See ‘Examples’ below.

Source

Law, M., Jackson, D., Turner, R., Rhodes, K., & Viechtbauer, W. (2016). Two new methods to fit models for network meta-analysis with random inconsistency effects. *BMC Medical Research Methodology*, **16**, 87. https://doi.org/10.1186/s12874-016-0184-5

Senn, S., Gavini, F., Magrez, D., & Scheen, A. (2013). Issues in performing a network meta-analysis. *Statistical Methods in Medical Research*, **22**(2), 169–189. https://doi.org/10.1177/0962280211432220

Examples

```
### copy data into 'dat' and examine data
dat <- dat.senn2013
dat

### create network graph ('igraph' package must be installed)
## Not run:
require(igraph)
pairs <- data.frame(do.call(rbind,
  sapply(split(dat$treatment, dat$study), function(x) t(combn(x,2)))), stringsAsFactors=FALSE)
pairs$X1 <- factor(pairs$X1, levels=sort(unique(dat$treatment)))
pairs$X2 <- factor(pairs$X2, levels=sort(unique(dat$treatment)))
tab <- table(pairs[,1], pairs[,2])
tab # adjacency matrix
g <- graph_from_adjacency_matrix(tab, mode = ”plus”, weighted=TRUE, diag=FALSE)
plot(g, edge.curved=FALSE, edge.width=E(g)$weight, layout=layout_as_star(g, center=”placebo”),
   vertex.size=45, vertex.color=”lightgray”, vertex.label.color=”black”, vertex.label.font=2)
## End(Not run)

### table of studies versus treatments examined
print(addmargins(table(dat$study, dat$treatment)), zero.print=””)

### table of frequencies with which treatment pairs were studied
print(as.table(crossprod(table(dat$study, dat$treatment))), zero.print=””)

### add means and sampling variances of the means to the dataset
dat <- escalc(measure=”MN”, mi=mi, sdi=sdi, ni=ni, data=dat)

### turn treatment variable into factor and set reference level
dat$treatment <- relevel(factor(dat$treatment), ref=”placebo”)

### add a space before each level (this makes the output a bit more legible)
levels(dat$treatment) <- paste0(” ”, levels(dat$treatment))

### network meta-analysis using an arm-based fixed-effects model with fixed study effects
res.fe <- rma.mv(yi, vi, mods = ~ study + treatment - 1, data=dat, slab=paste0(study, treatment))
res.fe

### test if treatment factor as a whole is significant
anova(res.fe, btt=”treatment”)

### forest plot of the contrast estimates (treatments versus placebos)
forest(tail(coef(res.fe), 9), tail(diag(vcov(res.fe)), 9), slab=levels(dat$treatment)[-1],
      xlim=c(-2.5, 2.0), alim=c(-1.5, 0.5), psize=1, xlab=”Estimate”, header=”Treatment”)

### weight matrix for the estimation of the fixed effects (leaving out the study effects)
w <- t(tail(vcov(res.fe) %*% t(model.matrix(res.fe)) %*% weights(res.fe, type=”matrix”), 9))
rownames(w) <- res.fe$slab

### create shade plot for the diabetes network with placebo as the reference treatment
### negative values in blue shades, positive values in red shades
cols <- colorRampPalette(c(”blue”, ”gray95”, ”red”))(9)
```

```
heatmap(w, Rowv=NA, Colv=NA, scale="none", margins=c(6,11), col=cols,
        cexRow=.7, cexCol=1, labCol=levels(dat$treatment)[-1])

### network meta-analysis using an arm-based random-effects model with fixed study effects
### by setting rho=1/2, tau^2 reflects the amount of heterogeneity for all treatment comparisons
res.re <- rma.mv(yi, vi, mods = ~ study + treatment - 1, random = ~ treatment | study, rho=1/2,
                 data=dat, slab=paste0(study, treatment))
res.re

### test if treatment factor as a whole is significant
anova(res.re, btt="treatment")

### forest plot of the contrast estimates (treatments versus placebos)
forest(tail(coef(res.re), 9), tail(diag(vcov(res.re)), 9), slab=levels(dat$treatment)[-1],
      xlim=c(-3.0, 2.5), alim=c(-1.5, 0.5), psize=1, xlab="Estimate", header="Treatment")

### compute the contribution of each study to the overall Q-test value
qi <- sort(by((resid(res.fe) / sqrt(dat$vi))^2, dat$study, sum))

### check that the values add up
sum(qi)
res.fe$QE

### plot the values
s <- length(qi)
par(mar=c(5,10,2,1))
plot(qi, 1:s, pch=19, xaxt="n", yaxt="n", xlim=c(0,40), xlab="Chi-Square Contribution", ylab="")
axis(side=1)
axis(side=2, at=1:s, labels=names(qi), las=1, tcl=0)
segments(rep(0,s), 1:s, qi, 1:s)

##############################################################################

### restructure dataset to a contrast-based format
dat <- dat.senn2013[c(1,4:2,5:6)] # reorder variables first
dat <- to.wide(dat, study="study", grp="treatment", ref="placebo", grpvars=4:6)
dat

### calculate mean difference and corresponding sampling variance for each treatment comparison
dat <- escalc(measure="MD", m1i=mi.1, sd1i=sdi.1, n1i=ni.1,
                             m2i=mi.2, sd2i=sdi.2, n2i=ni.2, data=dat)
dat

### calculate the variance-covariance matrix of the mean differences for the multitreatment studies
calc.v <- function(x) {
   v <- matrix(x$sdi.2[1]^2 / x$ni.2[1], nrow=nrow(x), ncol=nrow(x))
   diag(v) <- x$vi
   v
}
V <- bldiag(lapply(split(dat, dat$study), calc.v))

### add contrast matrix to dataset
dat <- contrmat(dat, grp1="treatment.1", grp2="treatment.2")
```

```
dat

### network meta-analysis using a contrast-based random-effects model
### by setting rho=1/2, tau^2 reflects the amount of heterogeneity for all treatment comparisons
### the treatment left out (placebo) becomes the reference level for the treatment comparisons
res <- rma.mv(yi, V, mods = ~ acarbose + benfluorex + metformin + miglitol + pioglitazone +
                             rosiglitazone + sitagliptin + sulfonylurea + vildagliptin - 1,
              random = ~ comp | study, rho=1/2, data=dat)
res

### estimate all pairwise differences between treatments (using the 'multcomp' package)
if (require(multcomp)) {
   contr <- contrMat(setNames(rep(1,res$p), colnames(res$X)), type="Tukey")
   sav <- predict(res, newmods=contr)
   sav[["slab"]] <- rownames(contr)
   sav
}

### fit random inconsistency effects model (see Law et al., 2016)
res <- rma.mv(yi, V, mods = ~ acarbose + benfluorex + metformin + miglitol + pioglitazone +
                             rosiglitazone + sitagliptin + sulfonylurea + vildagliptin - 1,
             random = list(~ comp | study, ~ comp | design), rho=1/2, phi=1/2, data=dat)
res
```

dat.vanhowe1999 *Studies on the Association between Circumcision and HIV Infection*

Description

Results from 33 studies examining the association between male circumcision and HIV infection.

Usage

```
dat.vanhowe1999
```

Format

The data frame contains the following columns:

study	`character`	study author
category	`character`	study type (high-risk group, partner study, or population survey)
non.pos	`numeric`	number of non-circumcised HIV positive cases
non.neg	`numeric`	number of non-circumcised HIV negative cases
cir.pos	`numeric`	number of circumcised HIV positive cases
cir.neg	`numeric`	number of circumcised HIV negative cases

Details

The 33 studies provide data in terms of 2×2 tables in the form:

	HIV positive	HIV negative
non-circumcised	`non.pos`	`non.neg`
circumcised	`cir.pos`	`cir.neg`

The goal of the meta-analysis was to examine if the risk of an HIV infection differs between non-circumcised versus circumcised men.

The dataset is interesting because it can be used to illustrate the difference between naively pooling results by summing up the counts across studies and then computing the odds ratio based on the aggregated table (as was done by Van Howe, 1999) and conducting a proper meta-analysis (as illustrated by O'Farrell & Egger, 2000). In fact, a proper meta-analysis shows that the HIV infection risk is on average higher in non-circumcised men, which is the opposite of what the naive pooling approach yields (which makes this an illustration of Simpson's paradox).

Source

Van Howe, R. S. (1999). Circumcision and HIV infection: Review of the literature and meta-analysis. *International Journal of STD & AIDS*, **10**(1), 8–16. https://doi.org/10.1258/0956462991913015

References

O'Farrell, N., & Egger, M. (2000). Circumcision in men and the prevention of HIV infection: A 'meta-analysis' revisited. *International Journal of STD & AIDS*, **11**(3), 137–142. https://doi.org/10.1258/0956462001915480

Examples

```
### copy data into 'dat'
dat <- dat.vanhowe1999

### naive pooling by summing up the counts within categories and then
### computing the odds ratios and corresponding confidence intervals
cat1 <- with(dat[dat$category=="high-risk group",],
  escalc(measure="OR", ai=sum(non.pos), bi=sum(non.neg), ci=sum(cir.pos), di=sum(cir.neg)))
cat2 <- with(dat[dat$category=="partner study",],
  escalc(measure="OR", ai=sum(non.pos), bi=sum(non.neg), ci=sum(cir.pos), di=sum(cir.neg)))
cat3 <- with(dat[dat$category=="population survey",],
  escalc(measure="OR", ai=sum(non.pos), bi=sum(non.neg), ci=sum(cir.pos), di=sum(cir.neg)))
summary(cat1, transf=exp, digits=2)
summary(cat2, transf=exp, digits=2)
summary(cat3, transf=exp, digits=2)

### naive pooling across all studies
all <- escalc(measure="OR", ai=sum(dat$non.pos), bi=sum(dat$non.neg),
                            ci=sum(dat$cir.pos), di=sum(dat$cir.neg))
summary(all, transf=exp, digits=2)

### calculate log odds ratios and corresponding sampling variances
dat <- escalc(measure="OR", ai=non.pos, bi=non.neg, ci=cir.pos, di=cir.neg, data=dat)
dat

### random-effects model
res <- rma(yi, vi, data=dat, method="DL")
```

```
res
predict(res, transf=exp, digits=2)

### random-effects model within subgroups
res <- rma(yi, vi, data=dat, method="DL", subset=category=="high-risk group")
predict(res, transf=exp, digits=2)
res <- rma(yi, vi, data=dat, method="DL", subset=category=="partner study")
predict(res, transf=exp, digits=2)
res <- rma(yi, vi, data=dat, method="DL", subset=category=="population survey")
predict(res, transf=exp, digits=2)
```

`dat.viechtbauer2021` *Studies to Illustrate Model Checking Methods*

Description

Results from 20 hypothetical randomized clinical trials examining the effectiveness of a medication for treating some disease.

Usage

```
dat.viechtbauer2021
```

Format

The data frame contains the following columns:

trial	`numeric`	trial number
nTi	`numeric`	number of patients in the treatment group
nCi	`numeric`	number of patients in the control group
xTi	`numeric`	number of patients in the treatment group with remission
xCi	`numeric`	number of patients in the control group with remission
dose	`numeric`	dosage of the medication provided to patients in the treatment group (in milligrams per day)

Details

The dataset was constructed for the purposes of illustrating the model checking and diagnostic methods described in Viechtbauer (2021). The code below provides the results for many of the analyses and plots discussed in the book chapter.

Source

Viechtbauer, W. (2021). Model checking in meta-analysis. In C. H. Schmid, T. Stijnen, & I. R. White (Eds.), *Handbook of meta-analysis* (pp. 219-254). Boca Raton, FL: CRC Press. https://doi.org/10.1201/978131511940

Examples

```
### copy data into 'dat'
```

```
dat <- dat.viechtbauer2021

### calculate log odds ratios and corresponding sampling variances

dat <- escalc(measure="OR", ai=xTi, n1i=nTi, ci=xCi, n2i=nCi, add=1/2, to="all", data=dat)
dat

### number of studies

k <- nrow(dat)

### fit models

res.CE <- rma(yi, vi, data=dat, method="FE")
res.CE

res.RE <- rma(yi, vi, data=dat, method="DL")
res.RE

res.MR <- rma(yi, vi, mods = ~ dose, data=dat, method="FE")
res.MR

res.ME <- rma(yi, vi, mods = ~ dose, data=dat, method="DL")
res.ME

### forest and bubble plot

par(mar=c(5,4,1,2))

forest(dat$yi, dat$vi, psize=0.8, efac=0, xlim=c(-4,6), ylim=c(-3,23),
       cex=1, width=c(5,5,5), xlab="Log Odds Ratio (LnOR)")
addpoly(res.CE, row=-1.5, cex=1, width=c(5,5,5), mlab="CE Model")
addpoly(res.RE, row=-2.5, cex=1, width=c(5,5,5), mlab="RE Model")
text(-4, 22, "Trial",         pos=4, font=2)
text( 6, 22, "LnOR [95% CI]", pos=2, font=2)
abline(h=0)

tmp <- regplot(res.ME, xlim=c(0,250), ylim=c(-1,1.5), predlim=c(0,250), shade=FALSE, digits=1,
               xlab="Dosage (mg per day)", psize="seinv", plim=c(NA,5), bty="l", las=1,
               lty=c("solid", "dashed"), label=TRUE, labsize=0.8, offset=c(1,0.7))
res.sub <- rma(yi, vi, mods = ~ dose, data=dat, method="DL", subset=-6)
abline(res.sub, lty="dotted")
points(tmp$xi, tmp$yi, pch=21, cex=tmp$psize, col="black", bg="darkgray")

par(mar=c(5,4,4,2))

### number of standardized deleted residuals larger than +-1.96 in each model

sum(abs(rstudent(res.CE)$z) >= qnorm(.975))
sum(abs(rstudent(res.MR)$z) >= qnorm(.975))
sum(abs(rstudent(res.RE)$z) >= qnorm(.975))
sum(abs(rstudent(res.ME)$z) >= qnorm(.975))
```

```
### plot of the standardized deleted residuals for the RE and ME models

plot(NA, NA, xlim=c(1,20), ylim=c(-4,4), xlab="Study", ylab="Standardized (Deleted) Residual",
     xaxt="n", main="Random-Effects Model", las=1)
axis(side=1, at=1:20)
abline(h=c(-1.96,1.96), lty="dotted")
abline(h=0)
points(1:20, rstandard(res.RE)$z, type="o", pch=19, col="gray70")
points(1:20, rstudent(res.RE)$z,  type="o", pch=19)
legend("top", pch=19, col=c("gray70","black"), lty="solid",
       legend=c("Standardized Residuals","Standardized Deleted Residuals"), bty="n")

plot(NA, NA, xlim=c(1,20), ylim=c(-4,4), xlab="Study", ylab="Standardized (Deleted) Residual",
     xaxt="n", main="Mixed-Effects Model", las=1)
axis(side=1, at=1:20)
abline(h=c(-1.96,1.96), lty="dotted")
abline(h=0)
points(1:20, rstandard(res.ME)$z, type="o", pch=19, col="gray70")
points(1:20, rstudent(res.ME)$z,  type="o", pch=19)
legend("top", pch=19, col=c("gray70","black"), lty="solid",
       legend=c("Standardized Residuals","Standardized Deleted Residuals"), bty="n")

### Baujat plots

baujat(res.CE, main="Common-Effects Model", xlab="Squared Pearson Residual", ylim=c(0,5), las=1)
baujat(res.ME, main="Mixed-Effects Model", ylim=c(0,2), las=1)

### GOSH plots (skipped because this takes quite some time to run)

if (FALSE) {

res.GOSH.CE <- gosh(res.CE, subsets=10^7)
plot(res.GOSH.CE, cex=0.2, out=6, xlim=c(-0.25,1.25), breaks=c(200,100))

res.GOSH.ME <- gosh(res.ME, subsets=10^7)
plot(res.GOSH.ME, het="tau2", out=6, breaks=50, adjust=0.6, las=1)

}

### plot of treatment dosage against the standardized residuals

plot(dat$dose, rstandard(res.ME)$z, pch=19, xlab="Dosage (mg per day)",
     ylab="Standardized Residual", xlim=c(0,250), ylim=c(-2.5,2.5), las=1)
abline(h=c(-1.96,1.96), lty="dotted", lwd=2)
abline(h=0)
title("Standardized Residual Plot")
text(dat$dose[6], rstandard(res.ME)$z[6], "6", pos=4, offset=0.4)

### quadratic polynomial model

rma(yi, vi, mods = ~ dose + I(dose^2), data=dat, method="DL")
```

```
### lack-of-fit model

resLOF <- rma(yi, vi, mods = ~ dose + factor(dose), data=dat, method="DL", btt=3:9)
resLOF

### scatter plot to illustrate the lack-of-fit model

regplot(res.ME, xlim=c(0,250), ylim=c(-1.0,1.5), xlab="Dosage (mg per day)", ci=FALSE,
       predlim=c(0,250), psize=1, pch=19, col="gray60", digits=1, lwd=1, bty="l", las=1)
dosages <- sort(unique(dat$dose))
lines(dosages, fitted(resLOF)[match(dosages, dat$dose)], type="o", pch=19, cex=2, lwd=2)
points(dat$dose, dat$yi, pch=19, col="gray60")
legend("bottomright", legend=c("Linear Model", "Lack-of-Fit Model"), pch=c(NA,19), col="black",
       lty="solid", lwd=c(1,2), pt.cex=c(1,2), seg.len=4, bty="n")

### checking normality of the standardized deleted residuals

qqnorm(res.ME, type="rstudent", main="Standardized Deleted Residuals", pch=19, label="out",
       lwd=2, pos=24, ylim=c(-4,3), lty=c("solid", "dotted"), las=1)

### checking normality of the random effects

sav <- qqnorm(ranef(res.ME)$pred, main="BLUPs of the Random Effects", cex=1, pch=19,
              xlim=c(-2.2,2.2), ylim=c(-0.6,0.6), las=1)
abline(a=0, b=sd(ranef(res.ME)$pred), lwd=2)
text(sav$x[6], sav$y[6], "6", pos=4, offset=0.4)

### hat values for the CE and RE models

plot(NA, NA, xlim=c(1,20), ylim=c(0,0.21), xaxt="n", las=1, xlab="Study", ylab="Hat Value")
axis(1, 1:20, cex.axis=1)
points(hatvalues(res.CE), type="o", pch=19, col="gray70")
points(hatvalues(res.RE), type="o", pch=19)
abline(h=1/20, lty="dotted", lwd=2)
title("Hat Values for the CE/RE Models")
legend("topright", pch=19, col=c("gray70","black"), lty="solid",
       legend=c("Common-Effects Model", "Random-Effects Model"), bty="n")

### heatmap of the hat matrix for the ME model

cols <- colorRampPalette(c("blue", "white", "red"))(101)
h <- hatvalues(res.ME, type="matrix")
image(1:nrow(h), 1:ncol(h), t(h[nrow(h):1,]), axes=FALSE,
    xlab="Influence of the Observed Effect of Study ...", ylab="On the Fitted Value of Study ...",
      col=cols, zlim=c(-max(abs(h)),max(abs(h))))
axis(1, 1:20, tick=FALSE)
axis(2, 1:20, labels=20:1, las=1, tick=FALSE)
abline(h=seq(0.5,20.5,by=1), col="white")
abline(v=seq(0.5,20.5,by=1), col="white")
points(1:20, 20:1, pch=19, cex=0.4)
title("Heatmap for the Mixed-Effects Model")

### plot of leverages versus standardized residuals for the ME model
```

```
plot(hatvalues(res.ME), rstudent(res.ME)$z, pch=19, cex=0.2+3*sqrt(cooks.distance(res.ME)),
     las=1, xlab="Leverage (Hat Value)", ylab="Standardized Deleted Residual",
     xlim=c(0,0.35), ylim=c(-3.5,2.5))
abline(h=c(-1.96,1.96), lty="dotted", lwd=2)
abline(h=0, lwd=2)
ids <- c(3,6,9)
text(hatvalues(res.ME)[ids] + c(0,0.013,0.010), rstudent(res.ME)$z[ids] - c(0.18,0,0), ids)
title("Leverage vs. Standardized Deleted Residuals")

### plot of the Cook's distances for the ME model

plot(1:20, cooks.distance(res.ME), ylim=c(0,1.6), type="o", pch=19, las=1, xaxt="n", yaxt="n",
     xlab="Study", ylab="Cook's Distance")
axis(1, 1:20, cex.axis=1)
axis(2, seq(0,1.6,by=0.4), las=1)
title("Cook's Distances")

### plot of the leave-one-out estimates of tau^2 for the ME model

x <- influence(res.ME)

plot(1:20, x$inf$tau2.del,  ylim=c(0,0.15), type="o", pch=19, las=1, xaxt="n", xlab="Study",
    ylab=expression(paste("Estimate of ", tau^2, " without the ", italic(i), "th study")))
abline(h=res.ME$tau2, lty="dashed")
axis(1, 1:20)
title("Residual Heterogeneity Estimates")

### plot of the covariance ratios for the ME model

plot(1:20, x$inf$cov.r,  ylim=c(0,2.0), type="o", pch=19, las=1, xaxt="n",
     xlab="Study", ylab="Covariance Ratio")
abline(h=1, lty="dashed")
axis(1, 1:20)
title("Covariance Ratios")

### fit mixed-effects model without studies 3 and/or 6

rma(yi, vi, mods = ~ dose, data=dat, method="DL", subset=-3)
rma(yi, vi, mods = ~ dose, data=dat, method="DL", subset=-6)
rma(yi, vi, mods = ~ dose, data=dat, method="DL", subset=-c(3,6))
```

dat.yusuf1985 *Studies of Beta Blockers During and After Myocardial Infarction*

Description

Results from studies examining the effectiveness of beta blockers for reducing mortality and reinfarction.

Usage

```
dat.yusuf1985
```

Format

The data frame contains the following columns:

table	`character`	table number
id	`character`	trial id number
trial	`character`	trial name or first author
ai	`numeric`	number of deaths/reinfarctions in treatment group
n1i	`numeric`	number of patients in treatment group
ci	`numeric`	number of deaths/reinfarctions in control group
n2i	`numeric`	number of patients in control group

Details

The dataset contains table 6 (total mortality from short-term trials of oral beta blockers), 9 (total mortality at one week from trials with an initial IV dose of a beta blocker), 10 (total mortality from long-term trials with treatment starting late and mortality from day 8 onwards in long-term trials that began early and continued after discharge), 11 (nonfatal reinfarction from long-term trials of beta blockers), 12a (sudden death in long-term beta blocker trials), and 12b (nonsudden death in long-term beta blocker trials) from the meta-analysis by Yusuf et al. (1985) on the effectiveness of of beta blockers for reducing mortality and reinfarction.

The article also describes what is sometimes called Peto's one-step method for meta-analyzing 2×2 table data. This method is implemented in the `rma.peto` function.

Source

Yusuf, S., Peto, R., Lewis, J., Collins, R., & Sleight, P. (1985). Beta blockade during and after myocardial infarction: An overview of the randomized trials. *Progress in Cardiovascular Disease*, **27**(5), 335–371. https://doi.org/10.1016/s0033-0620(85)80003-7

Examples

```
### copy data into 'dat'
dat <- dat.yusuf1985

### to select a table for the analysis
tab <- "6" ### either: 6, 9, 10, 11, 12a, 12b

### to double-check total counts as reported in article
apply(dat[dat$table==tab,4:7], 2, sum, na.rm=TRUE)

### meta-analysis using Peto's one-step method
res <- rma.peto(ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, subset=(table==tab))
res
predict(res, transf=exp, digits=2)
```

dfround *Round Variables in a Data Frame*

Description

Function to round the numeric variables in a data frame.

Usage

```
dfround(x, digits)
```

Arguments

x	a data frame.
digits	either a single integer or a numeric vector of the same length as there are columns in x.

Details

A simple convenience function to round the numeric variables in a data frame, possibly to different numbers of digits. Hence, digits can either be a single integer (which will then be used to round all numeric variables to the specified number of digits) or a numeric vector (of the same length as there are columns in x) to specify the number of digits to which each variable should be rounded.

Non-numeric variables are skipped. If digits is a vector, some arbitrary value (or NA) can be specified for those variables.

Value

Returns the data frame with variables rounded as specified.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

Examples

```
dat <- dat.bcg
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat)
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)
coef(summary(res))
dfround(coef(summary(res)), digits=c(2,3,2,3,2,2))
```

escalc *Calculate Effect Sizes and Outcome Measures*

Description

The function can be used to calculate various effect sizes or outcome measures (and the corresponding sampling variances) that are commonly used in meta-analyses.

Usage

```
escalc(measure, ai, bi, ci, di, n1i, n2i, x1i, x2i, t1i, t2i,
       m1i, m2i, sd1i, sd2i, xi, mi, ri, ti, sdi, r2i, ni, yi, vi, sei,
       data, slab, subset, include,
       add=1/2, to="only0", drop00=FALSE, vtype="LS",
       var.names=c("yi","vi"), add.measure=FALSE,
       append=TRUE, replace=TRUE, digits, ...)
```

Arguments

`measure`	a character string to specify which effect size or outcome measure should be calculated. See 'Details' for possible options and how the data needed to compute the selected effect size or outcome measure should then be specified.
`ai`	vector to specify the 2×2 table frequencies (upper left cell).
`bi`	vector to specify the 2×2 table frequencies (upper right cell).
`ci`	vector to specify the 2×2 table frequencies (lower left cell).
`di`	vector to specify the 2×2 table frequencies (lower right cell).
`n1i`	vector to specify the group sizes or row totals (first group/row).
`n2i`	vector to specify the group sizes or row totals (second group/row).
`x1i`	vector to specify the number of events (first group).
`x2i`	vector to specify the number of events (second group).
`t1i`	vector to specify the total person-times (first group).
`t2i`	vector to specify the total person-times (second group).
`m1i`	vector to specify the means (first group or time point).
`m2i`	vector to specify the means (second group or time point).
`sd1i`	vector to specify the standard deviations (first group or time point).
`sd2i`	vector to specify the standard deviations (second group or time point).
`xi`	vector to specify the frequencies of the event of interest.
`mi`	vector to specify the frequencies of the complement of the event of interest or the group means.
`ri`	vector to specify the raw correlation coefficients.
`ti`	vector to specify the total person-times.

sdi vector to specify the standard deviations.

r2i vector to specify the R^2 values.

ni vector to specify the sample/group sizes.

yi vector to specify the observed effect sizes or outcomes.

vi vector to specify the corresponding sampling variances.

sei vector to specify the corresponding standard errors.

data optional data frame containing the variables given to the arguments above.

slab optional vector with labels for the studies.

subset optional (logical or numeric) vector to specify the subset of studies that will be included in the data frame returned by the function.

include optional (logical or numeric) vector to specify the subset of studies for which the measure should be calculated. See the ‘Value’ section for more details.

add a non-negative number to specify the amount to add to zero cells, counts, or frequencies. See ‘Details’.

to a character string to specify when the values under add should be added (either "all", "only0", "if0all", or "none"). See ‘Details’.

drop00 logical to specify whether studies with no cases/events (or only cases) in both groups should be dropped when calculating the observed effect sizes or outcomes. See ‘Details’.

vtype a character string to specify the type of sampling variances to calculate. See ‘Details’.

var.names character string with two elements to specify the name of the variable for the observed effect sizes or outcomes and the name of the variable for the corresponding sampling variances (the defaults are "yi" and "vi").

add.measure logical to specify whether a variable should be added to the data frame (with default name "measure") that indicates the type of outcome measure computed. When using this option, var.names can have a third element to change this variable name.

append logical to specify whether the data frame provided via the data argument should be returned together with the observed effect sizes or outcomes and corresponding sampling variances (the default is TRUE).

replace logical to specify whether existing values for yi and vi in the data frame should be replaced or not. Only relevant when append=TRUE and the data frame already contains the yi and vi variables. If replace=TRUE (the default), all of the existing values will be overwritten. If replace=FALSE, only NA values will be replaced. See the ‘Value’ section for more details.

digits integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is 4. Note that the values are stored without rounding in the returned object.

... other arguments.

Details

Before a meta-analysis can be conducted, the relevant results from each study must be quantified in such a way that the resulting values can be further aggregated and compared. Depending on (a) the goals of the meta-analysis, (b) the design and types of studies included, and (c) the information provided therein, one of the various effect size or outcome measures described below may be appropriate for the meta-analysis and can be computed with the `escalc` function.

The `measure` argument is a character string to specify the outcome measure that should be calculated (see below for the various options), arguments `ai` through `ni` are then used to specify the information needed to calculate the various measures (depending on the chosen outcome measure, different arguments need to be specified), and `data` can be used to specify a data frame containing the variables given to the previous arguments. The `add`, `to`, and `drop00` arguments may be needed when dealing with frequency or count data that may need special handling when some of the frequencies or counts are equal to zero (see below for details). Finally, the `vtype` argument is used to specify how the sampling variances should be estimated (again, see below for details).

To provide a structure to the various effect size or outcome measures that can be calculated with the `escalc` function, we can distinguish between measures that are used to:

- contrast two independent (either experimentally created or naturally occurring) groups,
- describe the direction and strength of the association between two variables,
- summarize some characteristic or attribute of individual groups, or
- quantify change within a single group or the difference between two matched pairs samples.

Furthermore, where appropriate, we can further distinguish between measures that are applicable when the characteristic, response, or dependent variable assessed in the individual studies is:

- a dichotomous (binary) variable (e.g., remission versus no remission),
- a count of events per time unit (e.g., number of migraines per year),
- a quantitative variable (e.g., amount of depression as assessed by a rating scale).

Outcome Measures for Two-Group Comparisons:

In many meta-analyses, the goal is to synthesize the results from studies that compare or contrast two groups. The groups may be experimentally defined (e.g., a treatment and a control group created via random assignment) or may occur naturally (e.g., men and women, employees working under high- versus low-stress conditions, people exposed to some environmental risk factor versus those not exposed).

Measures for Dichotomous Variables:

In various fields (such as the health and medical sciences), the response variable measured is often dichotomous (binary), so that the data from a study comparing two different groups can be expressed in terms of a 2 × 2 table, such as:

	outcome 1	outcome 2	total
group 1	`ai`	`bi`	`n1i`
group 2	`ci`	`di`	`n2i`

where `ai`, `bi`, `ci`, and `di` denote the cell frequencies (i.e., the number of people falling into a particular category) and `n1i` and `n2i` are the row totals (i.e., the group sizes).

For example, in a set of randomized clinical trials, group 1 and group 2 may refer to the treatment and placebo/control group, respectively, with outcome 1 denoting some event of interest (e.g., death, complications, failure to improve under the treatment) and outcome 2 its complement. Similarly, in a set of cohort studies, group 1 and group 2 may denote those who engage in and those who do not engage in a potentially harmful behavior (e.g., smoking), with outcome 1 denoting the development of a particular disease (e.g., lung cancer) during the follow-up period. Finally, in a set of case-control studies, group 1 and group 2 may refer to those with the disease (i.e., cases) and those free of the disease (i.e., controls), with outcome 1 denoting, for example, exposure to some environmental risk factor in the past and outcome 2 non-exposure. Note that in all of these examples, the stratified sampling scheme fixes the row totals (i.e., the group sizes) by design.

A meta-analysis of studies reporting results in terms of 2×2 tables can be based on one of several different outcome measures, including the risk ratio (also called the relative risk), the odds ratio, the risk difference, and the arcsine square root transformed risk difference (e.g., Fleiss & Berlin, 2009, Rücker et al., 2009). For any of these outcome measures, one needs to specify the cell frequencies via the `ai`, `bi`, `ci`, and `di` arguments (or alternatively, one can use the `ai`, `ci`, `n1i`, and `n2i` arguments).

The options for the `measure` argument are then:

- `"RR"` for the *log risk ratio*,
- `"OR"` for the *log odds ratio*,
- `"RD"` for the *risk difference*,
- `"AS"` for the *arcsine square root transformed risk difference* (Rücker et al., 2009),
- `"PETO"` for the *log odds ratio* estimated with Peto's method (Yusuf et al., 1985).

Note that the log is taken of the risk ratio and the odds ratio, which makes these outcome measures symmetric around 0 and yields corresponding sampling distributions that are closer to normality. Also, when multiplied by 2, the arcsine square root transformed risk difference is actually identical to Cohen's h (Cohen, 1988).

Cell entries with a zero count can be problematic, especially for the risk ratio and the odds ratio. Adding a small constant to the cells of the 2×2 tables is a common solution to this problem. When `to="only0"` (the default), the value of `add` (the default is `1/2`; but see 'Note') is added to each cell of those 2×2 tables with at least one cell equal to 0. When `to="all"`, the value of `add` is added to each cell of all 2×2 tables. When `to="if0all"`, the value of `add` is added to each cell of all 2×2 tables, but only when there is at least one 2×2 table with a zero cell. Setting `to="none"` or `add=0` has the same effect: No adjustment to the observed table frequencies is made. Depending on the outcome measure and the data, this may lead to division by zero inside of the function (when this occurs, the resulting value is recoded to `NA`). Also, studies where `ai=ci=0` or `bi=di=0` may be considered to be uninformative about the size of the effect and dropping such studies has sometimes been recommended (Higgins et al., 2019). This can be done by setting `drop00=TRUE`. The values for such studies will then be set to `NA`.

Datasets corresponding to data of this type are provided in `dat.bcg`, `dat.collins1985a`, `dat.collins1985b`, `dat.egger2001`, `dat.hine1989`, `dat.laopaiboon2015`, `dat.lee2004`, `dat.li2007`, `dat.linde2005`, `dat.nielweise2007`, and `dat.yusuf1985`.

Assuming that the dichotomous outcome is actually a dichotomized version of the responses on an underlying quantitative scale, it is also possible to estimate the standardized mean difference based on 2×2 table data, using either the probit transformed risk difference or a transformation of the odds ratio (e.g., Cox & Snell, 1989; Chinn, 2000; Hasselblad & Hedges, 1995; Sánchez-Meca et al., 2003). The options for the `measure` argument are then:

- `"PBIT"` for the *probit transformed risk difference* as an estimate of the standardized mean difference,

- "OR2DN" for the *transformed odds ratio* as an estimate of the standardized mean difference (assuming normal distributions),
- "OR2DL" for the *transformed odds ratio* as an estimate of the standardized mean difference (assuming logistic distributions).

The probit transformation assumes that the responses on the underlying quantitative scale are normally distributed. There are two versions of the odds ratio transformation, the first also assuming normal distributions within the two groups, while the second assumes that the responses within groups follow logistic distributions.
A dataset corresponding to data of this type is provided in dat.gibson2002.

Measures for Event Counts:
In medical and epidemiological studies comparing two different groups (e.g., treated versus untreated patients, exposed versus unexposed individuals), results are sometimes reported in terms of event counts (i.e., the number of events, such as strokes or myocardial infarctions) over a certain period of time. Data of this type are also referred to as 'person-time data'. Assume that the studies report data in the form:

	number of events	total person-time
group 1	x1i	t1i
group 2	x2i	t2i

where x1i and x2i denote the number of events in the first and the second group, respectively, and t1i and t2i the corresponding total person-times at risk. Often, the person-time is measured in years, so that t1i and t2i denote the total number of follow-up years in the two groups.
This form of data is fundamentally different from what was described in the previous section, since the total follow-up time may differ even for groups of the same size and the individuals studied may experience the event of interest multiple times. Hence, different outcome measures than the ones described in the previous section need to be considered when data are reported in this format. These include the incidence rate ratio, the incidence rate difference, and the square root transformed incidence rate difference (Bagos & Nikolopoulos, 2009; Rothman et al., 2008). For any of these outcome measures, one needs to specify the total number of events via the x1i and x2i arguments and the corresponding total person-time values via the t1i and t2i arguments.
The options for the measure argument are then:

- "IRR" for the *log incidence rate ratio*,
- "IRD" for the *incidence rate difference*,
- "IRSD" for the *square root transformed incidence rate difference*.

Note that the log is taken of the incidence rate ratio, which makes this outcome measure symmetric around 0 and yields a corresponding sampling distribution that is closer to normality.
Studies with zero events in one or both groups can be problematic, especially for the incidence rate ratio. Adding a small constant to the number of events is a common solution to this problem. When to="only0" (the default), the value of add (the default is 1/2; but see 'Note') is added to x1i and x2i only in the studies that have zero events in one or both groups. When to="all", the value of add is added to x1i and x2i in all studies. When to="if0all", the value of add is added to x1i and x2i in all studies, but only when there is at least one study with zero events in one or both groups. Setting to="none" or add=0 has the same effect: No adjustment to the observed number of events is made. Depending on the outcome measure and the data, this may

lead to division by zero inside of the function (when this occurs, the resulting value is recoded to NA). Like for 2 × 2 table data, studies where x1i=x2i=0 may be considered to be uninformative about the size of the effect and dropping such studies has sometimes been recommended. This can be done by setting drop00=TRUE. The values for such studies will then be set to NA.

Datasets corresponding to data of this type are provided in dat.hart1999 and dat.nielweise2008.

Measures for Quantitative Variables:

When the response or dependent variable assessed in the individual studies is measured on some quantitative scale, it is customary to report certain summary statistics, such as the mean and standard deviation of the observations. The data layout for a study comparing two groups with respect to such a variable is then of the form:

	mean	standard deviation	group size
group 1	m1i	sd1i	n1i
group 2	m2i	sd2i	n2i

where m1i and m2i are the observed means of the two groups, sd1i and sd2i are the observed standard deviations, and n1i and n2i denote the number of individuals in each group. Again, the two groups may be experimentally created (e.g., a treatment and control group based on random assignment) or naturally occurring (e.g., men and women). In either case, the raw mean difference, the standardized mean difference, and the (log transformed) ratio of means (also called log response ratio) are useful outcome measures when meta-analyzing studies of this type.

The options for the measure argument are then:

- "MD" for the *raw mean difference* (e.g., Borenstein, 2009),
- "SMD" for the *standardized mean difference* (Hedges, 1981),
- "SMDH" for the *standardized mean difference* with heteroscedastic population variances in the two groups (Bonett, 2008, 2009),
- "ROM" for the *log transformed ratio of means* (Hedges et al., 1999; Lajeunesse, 2011).

For measure="ROM", the log is taken of the ratio of means, which makes this outcome measure symmetric around 0 and yields a corresponding sampling distribution that is closer to normality. Hence, this measure cannot be computed when m1i and m2i have opposite signs (i.e., it is meant to be used for ratio scale measurements, where both means should be positive anyway). For measure="SMD", the positive bias in the standardized mean difference is automatically corrected for within the function, yielding Hedges' g (Hedges, 1981). Similarly, the same bias correction is applied for measure="SMDH" (Bonett, 2009).

For measure="MD", one can choose between vtype="LS" (the default) and vtype="HO". The former computes the sampling variances without assuming homoscedasticity (i.e., that the true variances of the measurements are the same in group 1 and group 2 within each study), while the latter assumes homoscedasticity (equations 12.5 and 12.3 in Borenstein, 2009, respectively). For measure="SMD", one can choose between vtype="LS" (the default) for the usual large-sample approximation to compute the sampling variances (equation 8 in Hedges, 1982), vtype="UB" to compute unbiased estimates of the sampling variances (equation 9 in Hedges, 1983), vtype="LS2" to compute the sampling variances as described in Borenstein (2009) (i.e., equation 12.17), and vtype="AV" to compute the sampling variances with the usual large-sample approximation but plugging the sample-size weighted average of the Hedges' g values into the equation. For measure="ROM", one can choose between vtype="LS" (the default) for the usual large-sample approximation to compute the sampling variances (equation 1 in Hedges et al., 1999), vtype="HO" to compute the sampling variances assuming homoscedasticity (the unnumbered equation after equation 1 in Hedges et al., 1999), `vtype="AV"` to compute the sampling variances assuming homoscedasticity of the coefficient of variation within each group across studies, and `vtype="AVHO"` to compute the sampling variances assuming homoscedasticity of the coefficient of variation for both groups across studies.

Datasets corresponding to data of this type are provided in `dat.normand1999` and `dat.curtis1998`.

It is also possible to transform standardized mean differences into log odds ratios (e.g., Cox & Snell, 1989; Chinn, 2000; Hasselblad & Hedges, 1995; Sánchez-Meca et al., 2003). The options for the `measure` argument are then:

- `"D2ORN"` for the *transformed standardized mean difference* as an estimate of the log odds ratio (assuming normal distributions),
- `"D2ORL"` for the *transformed standardized mean difference* as an estimate of the log odds ratio (assuming logistic distributions).

Both of these transformations provide an estimate of the log odds ratio, the first assuming that the responses within the two groups are normally distributed, while the second assumes that the responses follow logistic distributions.

A dataset illustrating the combined analysis of standardized mean differences and probit transformed risk differences is provided in `dat.gibson2002`.

Finally, interest may also be focused on differences between the two groups with respect to their variability. Here, the (log transformed) ratio of the coefficient of variation of the two groups (also called the coefficient of variation ratio) can be a useful measure (Nakagawa et al., 2015). If focus is solely on the variability of the measurements within the two groups, then the (log transformed) ratio of the standard deviations (also called the variability ratio) can be used (Nakagawa et al., 2015). For the latter, one only needs to specify `sd1i`, `sd2i`, `n1i`, and `n2i`. The options for the `measure` argument are:

- `"CVR"` for the *log transformed coefficient of variation ratio*,
- `"VR"` for the *log transformed variability ratio*.

Note that a slight bias correction is applied for both of these measures (Nakagawa et al., 2015). Also, the sampling variance for `measure="CVR"` is computed as given by equation 12 in Nakagawa et al. (2015), but without the '$-2\rho\ldots$' terms, since for normally distributed data (which we assume here) the mean and variance (and transformations thereof) are independent.

Outcome Measures for Variable Association:

Meta-analyses are often used to synthesize studies that examine the direction and strength of the association between two variables measured concurrently and/or without manipulation by experimenters. In this section, a variety of outcome measures will be discussed that may be suitable for a meta-analyses with this purpose. We can distinguish between measures that are applicable when both variables are measured on quantitative scales, when both variables measured are dichotomous, and when the two variables are of mixed types.

Measures for Two Quantitative Variables:

The (Pearson or product-moment) correlation coefficient quantifies the direction and strength of the (linear) relationship between two quantitative variables and is therefore frequently used as the outcome measure for meta-analyses. Two alternative measures are a bias-corrected version of the correlation coefficient and Fisher's r-to-z transformed correlation coefficient.

For these measures, one needs to specify `ri`, the vector with the raw correlation coefficients, and `ni`, the corresponding sample sizes. The options for the `measure` argument are then:

- `"COR"` for the *raw correlation coefficient*,
- `"UCOR"` for the *raw correlation coefficient* corrected for its slight negative bias (based on equation 2.3 in Olkin & Pratt, 1958),

- "ZCOR" for *Fisher's r-to-z transformed correlation coefficient* (Fisher, 1921).

For measure="COR" and measure="UCOR", one can choose between vtype="LS" (the default) for the usual large-sample approximation to compute the sampling variances (i.e., plugging the (biased-corrected) correlation coefficients into equation 12.27 in Borenstein, 2009), vtype="UB" to compute unbiased estimates of the sampling variances (see Hedges, 1989, but using the exact equation instead of the approximation), and vtype="AV" to compute the sampling variances with the usual large-sample approximation but plugging the sample-size weighted average of the (bias-corrected) correlation coefficients into the equation.

Datasets corresponding to data of this type are provided in dat.mcdaniel1994 and dat.molloy2014.

For meta-analyses involving multiple correlations extracted from the same sample, see also the rcalc function.

Measures for Two Dichotomous Variables:

When the goal of a meta-analysis is to examine the relationship between two dichotomous variables, the data for each study can again be presented in the form of a 2 × 2 table, except that there may not be a clear distinction between the grouping variable and the outcome variable. Moreover, the table may be a result of cross-sectional (i.e., multinomial) sampling, where none of the table margins (except the total sample size) are fixed by the study design.

The phi coefficient and the odds ratio are commonly used measures of association for 2 × 2 table data (e.g., Fleiss & Berlin, 2009). The latter is particularly advantageous, as it is directly comparable to values obtained from stratified sampling (as described earlier). Yule's Q and Yule's Y (Yule, 1912) are additional measures of association for 2 × 2 table data (although they are not typically used in meta-analyses). Finally, assuming that the two dichotomous variables are actually dichotomized versions of the responses on two underlying quantitative scales (and assuming that the two variables follow a bivariate normal distribution), it is also possible to estimate the correlation between the two variables using the tetrachoric correlation coefficient (Pearson, 1900; Kirk, 1973).

For any of these outcome measures, one needs to specify the cell frequencies via the ai, bi, ci, and di arguments (or alternatively, one can use the ai, ci, n1i, and n2i arguments). The options for the measure argument are then:

- "OR" for the *log odds ratio*,
- "PHI" for the *phi coefficient*,
- "YUQ" for *Yule's Q* (Yule, 1912),
- "YUY" for *Yule's Y* (Yule, 1912),
- "RTET" for the *tetrachoric correlation coefficient*.

Tables with one or more zero counts are handled as described earlier. For measure="PHI", one must indicate via vtype="ST" or vtype="CS" whether the data for the studies were obtained using stratified or cross-sectional (i.e., multinomial) sampling, respectively (it is also possible to specify an entire vector for the vtype argument in case the sampling scheme differed for the various studies).

A dataset corresponding to data of this type is provided in dat.bourassa1996.

Measures for Mixed Variable Types:

Finally, we can consider outcome measures that can be used to describe the relationship between two variables, where one variable is dichotomous and the other variable measures some quantitative characteristic. In that case, it is likely that study authors again report summary statistics, such as the mean and standard deviation of the measurements within the two groups (defined by the dichotomous variable). Based on this information, one can compute the point-biserial correlation coefficient (Tate, 1954) as a measure of association between the two variables. If

the dichotomous variable is actually a dichotomized version of the responses on an underlying quantitative scale (and assuming that the two variables follow a bivariate normal distribution), it is also possible to estimate the correlation between the two variables using the biserial correlation coefficient (Pearson, 1909; Soper, 1914; Jacobs & Viechtbauer, 2017).
Here, one again needs to specify `m1i` and `m2i` for the observed means of the two groups, `sd1i` and `sd2i` for the observed standard deviations, and `n1i` and `n2i` for the number of individuals in each group. The options for the `measure` argument are then:

- `"RPB"` for the *point-biserial correlation coefficient*,
- `"RBIS"` for the *biserial correlation coefficient*.

For `measure="RPB"`, one must indicate via `vtype="ST"` or `vtype="CS"` whether the data for the studies were obtained using stratified or cross-sectional (i.e., multinomial) sampling, respectively (it is also possible to specify an entire vector for the `vtype` argument in case the sampling scheme differed for the various studies).

Outcome Measures for Individual Groups:

In this section, outcome measures will be described which may be useful when the goal of a meta-analysis is to synthesize studies that characterize some property of individual groups. We will again distinguish between measures that are applicable when the characteristic of interest is a dichotomous variable, when the characteristic represents an event count, or when the characteristic assessed is a quantitative variable.

Measures for Dichotomous Variables:
A meta-analysis may be conducted to aggregate studies that provide data about individual groups with respect to a dichotomous dependent variable. Here, one needs to specify `xi` and `ni`, denoting the number of individuals experiencing the event of interest and the total number of individuals within each study, respectively. Instead of specifying `ni`, one can use `mi` to specify the number of individuals that do not experience the event of interest. The options for the `measure` argument are then:

- `"PR"` for the *raw proportion*,
- `"PLN"` for the *log transformed proportion*,
- `"PLO"` for the *logit transformed proportion* (i.e., log odds),
- `"PAS"` for the *arcsine square root transformed proportion* (i.e., the angular transformation),
- `"PFT"` for the *Freeman-Tukey double arcsine transformed proportion* (Freeman & Tukey, 1950).

Zero cell entries can be problematic for certain outcome measures. When `to="only0"` (the default), the value of `add` (the default is `1/2`; but see ‘Note’) is added to `xi` and `mi` only for studies where `xi` or `mi` is equal to 0. When `to="all"`, the value of `add` is added to `xi` and `mi` in all studies. When `to="if0all"`, the value of `add` is added in all studies, but only when there is at least one study with a zero value for `xi` or `mi`. Setting `to="none"` or `add=0` has the same effect: No adjustment to the observed values is made. Depending on the outcome measure and the data, this may lead to division by zero inside of the function (when this occurs, the resulting value is recoded to `NA`).
Datasets corresponding to data of this type are provided in `dat.pritz1997` and `dat.debruin2009`.

Measures for Event Counts:
Various measures can be used to characterize individual groups when the dependent variable assessed is an event count. Here, one needs to specify `xi` and `ti`, denoting the number of events that occurred and the total person-times at risk, respectively. The options for the `measure` argument are then:

- "IR" for the *raw incidence rate*,
- "IRLN" for the *log transformed incidence rate*,
- "IRS" for the *square root transformed incidence rate*,
- "IRFT" for the *Freeman-Tukey transformed incidence rate* (Freeman & Tukey, 1950).

Measures "IR" and "IRLN" can also be used when meta-analyzing standardized incidence ratios (SIRs), where the observed number of events is divided by the expected number of events. In this case, arguments xi and ti are used to specify the observed and expected number of events in the studies. Since SIRs are not symmetric around 1, it is usually more appropriate to meta-analyze the log transformed SIRs (i.e., using measure "IRLN"), which are symmetric around 0.

Studies with zero events can be problematic, especially for the log transformed incidence rate. Adding a small constant to the number of events is a common solution to this problem. When to="only0" (the default), the value of add (the default is 1/2; but see 'Note') is added to xi only in the studies that have zero events. When to="all", the value of add is added to xi in all studies. When to="if0all", the value of add is added to xi in all studies, but only when there is at least one study with zero events. Setting to="none" or add=0 has the same effect: No adjustment to the observed number of events is made. Depending on the outcome measure and the data, this may lead to division by zero inside of the function (when this occurs, the resulting value is recoded to NA).

Measures for Quantitative Variables:

The goal of a meta-analysis may also be to characterize individual groups, where the response, characteristic, or dependent variable assessed in the individual studies is measured on some quantitative scale. In the simplest case, the raw mean for the quantitative variable is reported for each group, which then becomes the observed outcome for the meta-analysis. Here, one needs to specify mi, sdi, and ni for the observed means, the observed standard deviations, and the sample sizes, respectively. For ratio scale measurements, the log transformed mean or the log transformed coefficient of variation (with bias correction) may also be of interest (Nakagawa et al., 2015). If focus is solely on the variability of the measurements, then the log transformed standard deviation (with bias correction) is a useful measure (Nakagawa et al., 2015; Raudenbush & Bryk, 1987). Here, one only needs to specify sdi and ni.

The options for the measure argument are:

- "MN" for the *raw mean*,
- "MNLN" for the *log transformed mean*,
- "CVLN" for the *log transformed coefficient of variation*,
- "SDLN" for the *log transformed standard deviation*.

Note that sdi is used to specify the standard deviations of the observed values of the response, characteristic, or dependent variable and not the standard errors of the means. Also, the sampling variance for measure="CVLN" is computed as given by equation 27 in Nakagawa et al. (2015), but without the '$-2\rho\ldots$' term, since for normally distributed data (which we assume here) the mean and variance (and transformations thereof) are independent.

Outcome Measures for Change or Matched Pairs:

A more complicated situation arises when the purpose of the meta-analysis is to assess the amount of change within individual groups (e.g., before and after a treatment or under two different treatments) or when dealing with matched pairs designs.

Measures for Dichotomous Variables:

For dichotomous variables, the data for a study of this type gives rise to a paired 2×2 table, which is of the form:

	trt 2 outcome 1	trt 2 outcome 2
trt 1 outcome 1	`ai`	`bi`
trt 1 outcome 2	`ci`	`di`

where `ai`, `bi`, `ci`, and `di` denote the cell frequencies. Note that 'trt1' and 'trt2' may be applied to a single group of subjects or to matched pairs of subjects. Also, 'trt1' and 'trt2' might refer to two different time points (e.g., before and after a treatment). In any case, the data from such a study can be rearranged into a marginal table of the form:

	outcome 1	outcome 2
trt 1	`ai+bi`	`ci+di`
trt 2	`ai+ci`	`bi+di`

which is of the same form as a 2 × 2 table that would arise in a study comparing/contrasting two independent groups.

The options for the `measure` argument that will compute outcome measures based on the marginal table are:

- `"MPRR"` for the matched pairs *marginal log risk ratio*,
- `"MPOR"` for the matched pairs *marginal log odds ratio*,
- `"MPRD"` for the matched pairs *marginal risk difference*.

See Becker and Balagtas (1993), Curtin et al. (2002), Elbourne et al. (2002), Fagerland et al. (2014), May and Johnson (1997), Newcombe (1998), Stedman et al. (2011), and Zou (2007) for discussions of these measures.

The options for the `measure` argument that will compute outcome measures based on the paired table are:

- `"MPORC"` for the *conditional log odds ratio*,
- `"MPPETO"` for the *conditional log odds ratio* estimated with Peto's method.

See Curtin et al. (2002) and Zou (2007) for discussions of these measures.

Measures for Quantitative Variables:

When the response or dependent variable assessed in the individual studies is measured on some quantitative scale, the raw mean change, standardized versions thereof, or the (log transformed) ratio of means (log response ratio) can be used as outcome measures (Becker, 1988; Gibbons et al., 1993; Lajeunesse, 2011; Morris, 2000). Here, one needs to specify `m1i` and `m2i`, the observed means at the two measurement occasions, `sd1i` and `sd2i` for the corresponding observed standard deviations, `ri` for the correlation between the measurements at the two measurement occasions, and `ni` for the sample size. The options for the `measure` argument are then:

- `"MC"` for the *raw mean change*,
- `"SMCC"` for the *standardized mean change* using change score standardization (Gibbons et al., 1993),
- `"SMCR"` for the *standardized mean change* using raw score standardization (Becker, 1988),
- `"SMCRH"` for the *standardized mean change* using raw score standardization with heteroscedastic population variances at the two measurement occasions (Bonett, 2008),
- `"ROMC"` for the *log transformed ratio of means* (Lajeunesse, 2011).

See also Morris and DeShon (2002) for a thorough discussion of the difference between the change score measures.

A few notes about the change score measures. In practice, one often has a mix of information available from the individual studies to compute these measures. In particular, if `m1i` and `m2i` are unknown, but the raw mean change is directly reported in a particular study, then one can set m1i to that value and m2i to 0 (making sure that the raw mean change was computed as m1i-m2i within that study and not the other way around). Also, for the raw mean change ("MC") or the standardized mean change using change score standardization ("SMCC"), if sd1i, sd2i, and ri are unknown, but the standard deviation of the change scores is directly reported, then one can set sd1i to that value and both sd2i and ri to 0. Finally, for the standardized mean change using raw score standardization ("SMCR"), argument sd2i is actually not needed, as the standardization is only based on sd1i (Becker, 1988; Morris, 2000), which is usually the pre-test standard deviation (if the post-test standard deviation should be used, then set sd1i to that). Note that all of these measures are also applicable for matched-pairs designs (subscripts 1 and 2 then simply denote the first and second group that are formed by the matching).

Finally, interest may also be focused on differences in the variability of the measurements at the two measurement occasions (or between the two matched groups). Here, the (log transformed) ratio of the coefficient of variation (also called the coefficient of variation ratio) can be a useful measure (Nakagawa et al., 2015). If focus is solely on the variability of the measurements, then the (log transformed) ratio of the standard deviations (also called the variability ratio) can be used (Nakagawa et al., 2015). For the latter, one only needs to specify sd1i, sd2i, ni, and ri. The options for the measure argument are:

- "CVRC" for the *log transformed coefficient of variation ratio*,
- "VRC" for the *log transformed variability ratio*.

The definitions of these measures are the same as given in Nakagawa et al. (2015) but are here computed for two sets of dependent measurements. Hence, the computation of the sampling variances are adjusted to take the correlation between the measurements into consideration.

Other Outcome Measures for Meta-Analyses:

Other outcome measures are sometimes used for meta-analyses that do not directly fall into the categories above. These are described in this section.

Cronbach's alpha and Transformations Thereof:

Meta-analytic methods can also be used to aggregate Cronbach's alpha values from multiple studies. This is usually referred to as a 'reliability generalization meta-analysis' (Vacha-Haase, 1998). Here, one needs to specify ai, mi, and ni for the observed alpha values, the number of items/replications/parts of the measurement instrument, and the sample sizes, respectively. One can either directly analyze the raw Cronbach's alpha values or transformations thereof (Bonett, 2002, 2010; Hakstian & Whalen, 1976). The options for the measure argument are then:

- "ARAW" for *raw alpha* values,
- "AHW" for *transformed alpha values* (Hakstian & Whalen, 1976),
- "ABT" for *transformed alpha values* (Bonett, 2002).

Note that the transformations implemented here are slightly different from the ones described by Hakstian and Whalen (1976) and Bonett (2002). In particular, for "AHW", the transformation $1-(1-\alpha)^{1/3}$ is used, while for "ABT", the transformation $-\ln(1-\alpha)$ is used. This ensures that the transformed values are monotonically increasing functions of α.

A dataset corresponding to data of this type is provided in dat.bonett2010.

Partial and Semi-Partial Correlations:

Aloe and Becker (2012), Aloe and Thompson (2013), and Aloe (2014) describe the use of partial and semi-partial correlation coefficients as a method for meta-analyzing the results from regression models (when the focus is on a common regression coefficient of interest across studies). To compute these measures, one needs to specify ti for the test statistics (i.e., t-tests) of the regression coefficient of interest, ni for the sample sizes of the studies, mi for the number of predictors in the regression models, and r2i for the R^2 value of the regression models (the latter is only needed when measure="SPCOR"). The options for the measure argument are then:

- "PCOR" for the *partial correlation coefficient*,
- "ZPCOR" for *Fisher's r-to-z transformed partial correlation coefficient*,
- "SPCOR" for the *semi-partial correlation coefficient*.

Note that the sign of the (semi-)partial correlation coefficients is determined based on the signs of the values specified via the ti argument. Also, the Fisher transformation can only be applied to partial correlation coefficient, not semi-partial coefficients.

Converting a Data Frame to an 'escalc' Object:

The function can also be used to convert a regular data frame to an 'escalc' object. One simply sets the measure argument to one of the options described above (or to measure="GEN" for a generic outcome measure not further specified) and passes the observed effect sizes or outcomes via the yi argument and the corresponding sampling variances via the vi argument (or the standard errors via the sei argument).

Value

An object of class c("escalc","data.frame"). The object is a data frame containing the following components:

yi	observed effect sizes or outcomes.
vi	corresponding sampling variances.

If append=TRUE and a data frame was specified via the data argument, then yi and vi are appended to this data frame. Note that the var.names argument actually specifies the names of these two variables (yi and vi are the defaults).

If the data frame already contains two variables with names as specified by the var.names argument, the values for these two variables will be overwritten when replace=TRUE (which is the default). By setting replace=FALSE, only values that are NA will be replaced.

The subset argument can be used to select the studies that will be included in the data frame returned by the function. On the other hand, the include argument simply selects for which studies the measure will be computed (if it shouldn't be computed for all of them).

The object is formatted and printed with the print.escalc function. The summary.escalc function can be used to obtain confidence intervals for the individual outcomes.

With the aggregate.escalc function, one can aggregate multiple effect sizes or outcomes belonging to the same study (or some other clustering variable) into a single combined effect size or outcome.

Note

The variable names specified under var.names should be syntactically valid variable names. If necessary, they are adjusted so that they are.

Although the default value for add is 1/2, for certain measures the use of such a bias correction makes little sense and for these measures, the function internally sets add=0. This applies to the following measures: "AS", "PHI", "RTET", "IRSD", "PAS", "PFT", "IRS", and "IRFT". One can still force the use of the bias correction by explicitly setting the add argument to some non-zero value.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Aloe, A. M. (2014). An empirical investigation of partial effect sizes in meta-analysis of correlational data. *Journal of General Psychology*, **141**(1), 47–64. https://doi.org/10.1080/00221309.2013.853021

Aloe, A. M., & Becker, B. J. (2012). An effect size for regression predictors in meta-analysis. *Journal of Educational and Behavioral Statistics*, **37**(2), 278–297. https://doi.org/10.3102/1076998610396901

Aloe, A. M., & Thompson, C. G. (2013). The synthesis of partial effect sizes. *Journal of the Society for Social Work and Research*, **4**(4), 390–405. https://doi.org/10.5243/jsswr.2013.24

Bagos, P. G., & Nikolopoulos, G. K. (2009). Mixed-effects Poisson regression models for meta-analysis of follow-up studies with constant or varying durations. *The International Journal of Biostatistics*, **5**(1). https://doi.org/10.2202/1557-4679.1168

Becker, B. J. (1988). Synthesizing standardized mean-change measures. *British Journal of Mathematical and Statistical Psychology*, **41**(2), 257–278. https://doi.org/10.1111/j.2044-8317.1988.tb00901.x

Becker, M. P., & Balagtas, C. C. (1993). Marginal modeling of binary cross-over data. *Biometrics*, **49**(4), 997–1009. https://doi.org/10.2307/2532242

Bonett, D. G. (2002). Sample size requirements for testing and estimating coefficient alpha. *Journal of Educational and Behavioral Statistics*, **27**(4), 335–340. https://doi.org/10.3102/10769986027004335

Bonett, D. G. (2008). Confidence intervals for standardized linear contrasts of means. *Psychological Methods*, **13**(2), 99–109. https://doi.org/10.1037/1082-989X.13.2.99

Bonett, D. G. (2009). Meta-analytic interval estimation for standardized and unstandardized mean differences. *Psychological Methods*, **14**(3), 225–238. https://doi.org/10.1037/a0016619

Bonett, D. G. (2010). Varying coefficient meta-analytic methods for alpha reliability. *Psychological Methods*, **15**(4), 368–385. https://doi.org/10.1037/a0020142

Borenstein, M. (2009). Effect sizes for continuous data. In H. Cooper, L. V. Hedges, & J. C. Valentine (Eds.), *The handbook of research synthesis and meta-analysis* (2nd ed., pp. 221–235). New York: Russell Sage Foundation.

Chinn, S. (2000). A simple method for converting an odds ratio to effect size for use in meta-analysis. *Statistics in Medicine*, **19**(22), 3127–3131. https://doi.org/10.1002/1097-0258(20001130)19:22<3127::aid-sim784>3.0.co;2-m

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

Cox, D. R., & Snell, E. J. (1989). *Analysis of binary data* (2nd ed.). London: Chapman & Hall.

Curtin, F., Elbourne, D., & Altman, D. G. (2002). Meta-analysis combining parallel and cross-over clinical trials. II: Binary outcomes. *Statistics in Medicine*, **21**(15), 2145–2159. https://doi.org/10.1002/sim.1206

Elbourne, D. R., Altman, D. G., Higgins, J. P. T., Curtin, F., Worthington, H. V., & Vail, A. (2002). Meta-analyses involving cross-over trials: Methodological issues. *International Journal of Epidemiology*, **31**(1), 140–149. https://doi.org/10.1093/ije/31.1.140

Fagerland, M. W., Lydersen, S., & Laake, P. (2014). Recommended tests and confidence intervals for paired binomial proportions. *Statistics in Medicine*, **33**(16), 2850–2875. https://doi.org/10.1002/sim.6148

Fisher, R. A. (1921). On the “probable error” of a coefficient of correlation deduced from a small sample. *Metron*, **1**, 1–32. http://hdl.handle.net/2440/15169

Fleiss, J. L., & Berlin, J. (2009). Effect sizes for dichotomous data. In H. Cooper, L. V. Hedges, & J. C. Valentine (Eds.), *The handbook of research synthesis and meta-analysis* (2nd ed., pp. 237–253). New York: Russell Sage Foundation.

Freeman, M. F., & Tukey, J. W. (1950). Transformations related to the angular and the square root. *Annals of Mathematical Statistics*, **21**(4), 607–611. https://doi.org/10.1214/aoms/1177729756

Gibbons, R. D., Hedeker, D. R., & Davis, J. M. (1993). Estimation of effect size from a series of experiments involving paired comparisons. *Journal of Educational Statistics*, **18**(3), 271–279. https://doi.org/10.3102/10769986018003271

Hakstian, A. R., & Whalen, T. E. (1976). A k-sample significance test for independent alpha coefficients. *Psychometrika*, **41**(2), 219–231. https://doi.org/10.1007/BF02291840

Hasselblad, V., & Hedges, L. V. (1995). Meta-analysis of screening and diagnostic tests. Psychological Bulletin, 117(1), 167-178. https://doi.org/10.1037/0033-2909.117.1.167

Hedges, L. V. (1981). Distribution theory for Glass’s estimator of effect size and related estimators. *Journal of Educational Statistics*, **6**(2), 107–128. https://doi.org/10.3102/10769986006002107

Hedges, L. V. (1982). Estimation of effect size from a series of independent experiments. *Psychological Bulletin*, **92**(2), 490–499. https://doi.org/10.1037/0033-2909.92.2.490

Hedges, L. V. (1983). A random effects model for effect sizes. *Psychological Bulletin*, **93**(2), 388–395. https://doi.org/10.1037/0033-2909.93.2.388

Hedges, L. V. (1989). An unbiased correction for sampling error in validity generalization studies. *Journal of Applied Psychology*, **74**(3), 469–477. https://doi.org/10.1037/0021-9010.74.3.469

Hedges, L. V., Gurevitch, J., & Curtis, P. S. (1999). The meta-analysis of response ratios in experimental ecology. *Ecology*, **80**(4), 1150–1156. https://doi.org/10.1890/0012-9658(1999)080[1150:TMAORR]2.0.CO;2

Higgins, J. P. T., Thomas, J., Chandler, J., Cumpston, M., Li, T., Page, M. J., & Welch, V. A. (Eds.) (2019). *Cochrane handbook for systematic reviews of interventions* (2nd ed.). Chichester, UK: Wiley. https://training.cochrane.org/handbook

Jacobs, P., & Viechtbauer, W. (2017). Estimation of the biserial correlation and its sampling variance for use in meta-analysis. *Research Synthesis Methods*, **8**(2), 161–180. https://doi.org/10.1002/jrsm.1218

Kirk, D. B. (1973). On the numerical approximation of the bivariate normal (tetrachoric) correlation coefficient. *Psychometrika*, **38**(2), 259–268. https://doi.org/10.1007/BF02291118

Lajeunesse, M. J. (2011). On the meta-analysis of response ratios for studies with correlated and multi-group designs. *Ecology*, **92**(11), 2049–2055. https://doi.org/10.1890/11-0423.1

May, W. L., & Johnson, W. D. (1997). Confidence intervals for differences in correlated binary proportions. *Statistics in Medicine*, **16**(18), 2127–2136. https://doi.org/10.1002/(SICI)1097-0258(19970930)16:18<2127::AID-SIM633>3.0.CO;2-W

Morris, S. B. (2000). Distribution of the standardized mean change effect size for meta-analysis on repeated measures. *British Journal of Mathematical and Statistical Psychology*, **53**(1), 17–29. https://doi.org/10.1348/000711000159150

Morris, S. B., & DeShon, R. P. (2002). Combining effect size estimates in meta-analysis with repeated measures and independent-groups designs. *Psychological Methods*, **7**(1), 105–125. https://doi.org/10.1037/1082-989x.7.1.105

Nakagawa, S., Poulin, R., Mengersen, K., Reinhold, K., Engqvist, L., Lagisz, M., & Senior, A. M. (2015). Meta-analysis of variation: Ecological and evolutionary applications and beyond. *Methods in Ecology and Evolution*, **6**(2), 143–152. https://doi.org/10.1111/2041-210x.12309

Newcombe, R. G. (1998). Improved confidence intervals for the difference between binomial proportions based on paired data. *Statistics in Medicine*, **17**(22), 2635–2650. https://doi.org/10.1002/(SICI)1097-0258(19981130)17:22<2635::AID-SIM954>3.0.CO;2-C

Olkin, I., & Pratt, J. W. (1958). Unbiased estimation of certain correlation coefficients. *Annals of Mathematical Statistics*, **29**(1), 201–211. https://doi.org/10.1214/aoms/1177706717

Pearson, K. (1900). Mathematical contributions to the theory of evolution. VII. On the correlation of characters not quantitatively measurable. *Philosophical Transactions of the Royal Society of London, Series A*, **195**, 1–47. https://doi.org/10.1098/rsta.1900.0022

Pearson, K. (1909). On a new method of determining correlation between a measured character A, and a character B, of which only the percentage of cases wherein B exceeds (or falls short of) a given intensity is recorded for each grade of A. *Biometrika*, **7**(1/2), 96–105. https://doi.org/10.1093/biomet/7.1-2.96

Raudenbush, S. W., & Bryk, A. S. (1987). Examining correlates of diversity. *Journal of Educational Statistics*, **12**(3), 241–269. https://doi.org/10.3102/10769986012003241

Rothman, K. J., Greenland, S., & Lash, T. L. (2008). *Modern epidemiology* (3rd ed.). Philadelphia: Lippincott Williams & Wilkins.

Rücker, G., Schwarzer, G., Carpenter, J., & Olkin, I. (2009). Why add anything to nothing? The arcsine difference as a measure of treatment effect in meta-analysis with zero cells. *Statistics in Medicine*, **28**(5), 721–738. https://doi.org/10.1002/sim.3511

Sánchez-Meca, J., Marín-Martínez, F., & Chacón-Moscoso, S. (2003). Effect-size indices for dichotomized outcomes in meta-analysis. *Psychological Methods*, **8**(4), 448–467. https://doi.org/10.1037/1082-989X.8.4.448

Soper, H. E. (1914). On the probable error of the bi-serial expression for the correlation coefficient. *Biometrika*, **10**(2/3), 384–390. https://doi.org/10.1093/biomet/10.2-3.384

Stedman, M. R., Curtin, F., Elbourne, D. R., Kesselheim, A. S., & Brookhart, M. A. (2011). Meta-analyses involving cross-over trials: Methodological issues. *International Journal of Epidemiology*, **40**(6), 1732–1734. https://doi.org/10.1093/ije/dyp345

Tate, R. F. (1954). Correlation between a discrete and a continuous variable: Point-biserial correlation. *Annals of Mathematical Statistics*, **25**(3), 603–607. https://doi.org/10.1214/aoms/1177728730

Vacha-Haase, T. (1998). Reliability generalization: Exploring variance in measurement error affecting score reliability across studies. *Educational and Psychological Measurement*, **58**(1), 6–20. https://doi.org/10.1177/0013164498058001002

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Yule, G. U. (1912). On the methods of measuring association between two attributes. *Journal of the Royal Statistical Society*, **75**(6), 579–652. https://doi.org/10.2307/2340126

Yusuf, S., Peto, R., Lewis, J., Collins, R., & Sleight, P. (1985). Beta blockade during and after myocardial infarction: An overview of the randomized trials. *Progress in Cardiovascular Disease*, **27**(5), 335–371. https://doi.org/10.1016/s0033-0620(85)80003-7

Zou, G. Y. (2007). One relative risk versus two odds ratios: Implications for meta-analyses involving paired and unpaired binary data. *Clinical Trials*, **4**(1), 25–31. https://doi.org/10.1177/1740774506075667

See Also

print.escalc, summary.escalc, rma.uni, rma.mh, rma.peto, rma.glmm, rma.mv

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)
dat

### suppose that for a particular study, yi and vi are known (i.e., have
### already been calculated) but the 2x2 table counts are not known; with
### replace=FALSE, the yi and vi values for that study are not replaced
dat[1:12,10:11] <- NA
dat[13,4:7] <- NA
dat
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat, replace=FALSE)
dat

### illustrate difference between 'subset' and 'include' arguments
escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, subset=1:6)
escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, include=1:6)

### convert a regular data frame to an 'escalc' object
### dataset from Lipsey & Wilson (2001), Table 7.1, page 130
dat <- data.frame(id = c(100, 308, 1596, 2479, 9021, 9028, 161, 172, 537, 7049),
                  yi = c(-0.33, 0.32, 0.39, 0.31, 0.17, 0.64, -0.33, 0.15, -0.02, 0.00),
                  vi = c(0.084, 0.035, 0.017, 0.034, 0.072, 0.117, 0.102, 0.093, 0.012, 0.067),
                  random = c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1),
                  intensity = c(7, 3, 7, 5, 7, 7, 4, 4, 5, 6))
dat <- escalc(measure="SMD", yi=yi, vi=vi, data=dat, slab=paste("Study ID:", id), digits=3)
dat
```

---

fitstats *Fit Statistics and Information Criteria for 'rma' Objects*

Description

Functions to extract the log-likelihood, deviance, AIC, BIC, and AICc values from objects of class "rma".

Usage

```
fitstats(object, ...)

## S3 method for class 'rma'
fitstats(object, ..., REML)

## S3 method for class 'rma'
logLik(object, REML, ...)
```

```
## S3 method for class 'rma'
deviance(object, REML, ...)

## S3 method for class 'rma'
AIC(object, ..., k=2, correct=FALSE)
## S3 method for class 'rma'
BIC(object, ...)
```

Arguments

`object`	an object of class `"rma"`.
`...`	optionally more fitted model objects.
`REML`	logical to specify whether the regular or restricted likelihood function should be used to obtain the fit statistics and information criteria. Defaults to the method of estimation used, that is `TRUE` if `object` was fitted with `method="REML"` and `FALSE` otherwise.
`k`	numeric value to specify the penalty per parameter to use. The default (`k=2`) is the classical AIC. See `AIC` for more details.
`correct`	logical to specify whether the regular (default) or corrected (i.e., AICc) should be extracted.

Value

For `fitstats`, a data frame with the (restricted) log-likelihood, deviance, AIC, BIC, and AICc values for each model passed to the function.

For `logLik`, an object of class `"logLik"`, providing the (restricted) log-likelihood of the model evaluated at the estimated coefficient(s).

For `deviance`, a numeric value with the corresponding deviance.

For `AIC` and `BIC`, either a numeric value with the corresponding AIC, AICc, or BIC or a data frame with rows corresponding to the models and columns representing the number of parameters in the model (`df`) and the AIC, AICc, or BIC.

Note

Variance components in the model (e.g., τ^2 in random/mixed-effects models fitted with `rma.uni`) are counted as additional parameters in the calculation of the AIC, BIC, and AICc. Also, the fixed effects are counted as parameters in the calculation of the AIC, BIC, and AICc even when using REML estimation.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mh, rma.peto, rma.glmm, rma.mv, anova.rma

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### random-effects model
res1 <- rma(yi, vi, data=dat, method="ML")

### mixed-effects model with absolute latitude and publication year as moderators
res2 <- rma(yi, vi, mods = ~ ablat + year, data=dat, method="ML")

### compare fit statistics
fitstats(res1, res2)

### log-likelihoods
logLik(res1)
logLik(res2)

### deviances
deviance(res1)
deviance(res2)

### AIC, AICc, and BIC values
AIC(res1, res2)
AIC(res1, res2, correct=TRUE)
BIC(res1, res2)
```

---

fitted.rma *Fitted Values for 'rma' Objects*

Description

The function computes the fitted values for objects of class "rma".

Usage

```
## S3 method for class 'rma'
fitted(object, ...)
```

Arguments

object	an object of class "rma".
...	other arguments.

Value

A vector with the fitted values.

Note

The predict.rma function also provides standard errors and confidence intervals for the fitted values. Best linear unbiased predictions (BLUPs) that combine the fitted values based on the fixed effects and the estimated contributions of the random effects can be obtained with blup.rma.uni (only for objects of class "rma.uni").

For objects not involving moderators, the fitted values are all identical to the estimated value of the model intercept.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

predict.rma, blup.rma.uni

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)

### compute the fitted values
fitted(res)
```

---

forest *Forest Plots*

Description

The forest function can be used to create forest plots.

Usage

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

Arguments

x either an object of class "rma", a vector with the observed effect sizes or outcomes, or an object of class "cumul.rma". See ‘Details’.

... other arguments.

Details

Currently, methods exist for three types of situations.

In the first case, object `x` is a fitted model object coming from the `rma.uni`, `rma.mh`, or `rma.peto` functions. The corresponding method is then `forest.rma`.

Alternatively, object `x` can be a vector with observed effect sizes or outcomes. The corresponding method is then `forest.default`.

Finally, object `x` can be an object coming from the `cumul.rma.uni`, `cumul.rma.mh`, or `cumul.rma.peto` functions. The corresponding method is then `forest.cumul.rma`.

Author(s)

Wolfgang Viechtbauer <`wvb@metafor-project.org`> `https://www.metafor-project.org`

References

Lewis, S., & Clarke, M. (2001). Forest plots: Trying to see the wood and the trees. *British Medical Journal*, **322**(7300), 1479–1480. https://doi.org/10.1136/bmj.322.7300.1479

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`forest.rma`, `forest.default`, `forest.cumul.rma`

`forest.cumul.rma` *Forest Plots (Method for 'cumul.rma' Objects)*

Description

Function to create forest plots for objects of class `"cumul.rma"`.

Usage

```
## S3 method for class 'cumul.rma'
forest(x, annotate=TRUE, header=FALSE,
       xlim, alim, olim, ylim, top=3, at, steps=5,
       level=x$level, refline=0, digits=2L, width,
       xlab, ilab, ilab.xpos, ilab.pos,
       transf, atransf, targs, rows,
       efac=1, pch=15, psize=1, col,
       lty, fonts, cex, cex.lab, cex.axis, annosym, ...)
```

Arguments

`x`	an object of class `"cumul.rma"`.
`annotate`	logical to specify whether annotations should be added to the plot (the default is `TRUE`).
`header`	logical to specify whether column headings should be added to the plot (the default is `FALSE`). Can also be a character vector to specify the left and right headings.
`xlim`	horizontal limits of the plot region. If unspecified, the function tries to set the horizontal plot limits to some sensible values.
`alim`	the actual x-axis limits. If unspecified, the function tries to set the x-axis limits to some sensible values.
`olim`	optional argument to specify observation/outcome limits. If unspecified, no limits are used.
`ylim`	the y-axis limits of the plot. If unspecified, the function tries to set the y-axis limits to some sensible values.
`top`	the amount of space to leave empty at the top of the plot (e.g., for adding headers) (the default is 3 rows).
`at`	position of the x-axis tick marks and corresponding labels. If unspecified, the function tries to set the tick mark positions/labels to some sensible values.
`steps`	the number of tick marks for the x-axis (the default is 5). Ignored when the positions are specified via the `at` argument.
`level`	numeric value between 0 and 100 to specify the confidence interval level (the default is to take the value from the object).
`refline`	numeric value to specify the location of the vertical ‘reference’ line (the default is 0). The line can be suppressed by setting this argument to `NA`.
`digits`	integer to specify the number of decimal places to which the tick mark labels of the x-axis and the annotations should be rounded (the default is `2L`). Can also be a vector of two integers, the first to specify the number of decimal places for the annotations, the second for the x-axis labels. When specifying an integer (e.g., `2L`), trailing zeros after the decimal mark are dropped for the x-axis labels. When specifying a numeric value (e.g., `2`), trailing zeros are retained.
`width`	optional integer to manually adjust the width of the columns for the annotations.
`xlab`	title for the x-axis. If unspecified, the function tries to set an appropriate axis title.
`ilab`	optional vector, matrix, or data frame providing additional information about the studies that should be added to the plot.
`ilab.xpos`	numeric vector to specify the x-axis position(s) of the variable(s) given via `ilab` (must be specified if `ilab` is specified).
`ilab.pos`	integer(s) (either 1, 2, 3, or 4) to specify the alignment of the vector(s) given via `ilab` (2 means right, 4 mean left aligned). If unspecified, the default is to center the labels.

`transf` optional argument to specify the name of a function that should be used to transform the estimates and confidence interval bounds (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`atransf` optional argument to specify the name of a function that should be used to transform the x-axis labels and annotations (e.g., `atransf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified via `transf` or `atransf`.

`rows` optional vector to specify the rows (or more generally, the horizontal positions) for plotting the outcomes. Can also be a single value to specify the row (horizontal position) of the first outcome (the remaining outcomes are then plotted below this starting row). If unspecified, the function sets this value automatically.

`efac` vertical expansion factor for confidence interval limits and arrows. The default value of 1 should usually work okay. Can also be a vector of two numbers, the first for CI limits, the second for arrows.

`pch` plotting symbol to use for the estimates. By default, a filled square is used. See `points` for other options. Can also be a vector of values.

`psize` numeric value to specify the point sizes for the estimates (the default is 1). Can also be a vector of values.

`col` optional character string to specify the name of a color to use for plotting (`"black"` is used by default if not specified). Can also be a vector of color names.

`lty` optional character string to specify the line type for the confidence intervals. If unspecified, the function sets this to `"solid"` by default.

`fonts` optional character string to specify the font to use for the study labels, annotations, and the extra information (if specified via `ilab`). If unspecified, the default font is used.

`cex` optional character and symbol expansion factor. If unspecified, the function tries to set this to a sensible value.

`cex.lab` optional expansion factor for the x-axis title. If unspecified, the function tries to set this to a sensible value.

`cex.axis` optional expansion factor for the x-axis labels. If unspecified, the function tries to set this to a sensible value.

`annosym` optional vector of length 3 to change the left bracket, separation, and right bracket symbols for the annotations.

`...` other arguments.

Details

The plot shows the estimated (average) outcome with corresponding confidence interval as one study at a time is added to the analysis.

Note

The function tries to set some sensible values for the optional arguments, but it may be necessary to adjust these in certain circumstances.

The function actually returns some information about the chosen defaults invisibly. Printing this information is useful as a starting point to make adjustments to the plot.

If the number of studies is quite large, the labels, annotations, and symbols may become quite small and impossible to read. Stretching the plot window vertically may then provide a more readable figure (one should call the function again after adjusting the window size, so that the label/symbol sizes can be properly adjusted). Also, the `cex`, `cex.lab`, and `cex.axis` arguments are then useful to adjust the symbol and text sizes.

If the horizontal plot and/or x-axis limits are set manually, then the horizontal plot limits (`xlim`) must be at least as wide as the x-axis limits (`alim`). This restriction is enforced inside the function.

If the outcome measure used for creating the plot is bounded (e.g., correlations are bounded between -1 and +1, proportions are bounded between 0 and 1), one can use the `olim` argument to enforce those limits (the observed outcomes and confidence intervals cannot exceed those bounds then).

The `lty` argument can also be a vector of two elements, the first for specifying the line type of the individual CIs (`"solid"` by default), the second for the line type of the horizontal line that is automatically added to the plot (`"solid"` by default; set to `"blank"` to remove it).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Chalmers, T. C., & Lau, J. (1993). Meta-analytic stimulus for changes in clinical trials. *Statistical Methods in Medical Research*, **2**(2), 161–172. https://doi.org/10.1177/096228029300200204

Lau, J., Schmid, C. H., & Chalmers, T. C. (1995). Cumulative meta-analysis of clinical trials builds evidence for exemplary medical care. *Journal of Clinical Epidemiology*, **48**(1), 45–57. https://doi.org/10.1016/0895-4356(94)00106-z

Lewis, S., & Clarke, M. (2001). Forest plots: Trying to see the wood and the trees. *British Medical Journal*, **322**(7300), 1479–1480. https://doi.org/10.1136/bmj.322.7300.1479

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`forest`, `cumul`

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit random-effects model
res <- rma(yi, vi, data=dat, slab=paste(author, year, sep=", "))

### draw cumulative forest plots
x <- cumul(res, order=dat$year)
forest(x, cex=.8, header=TRUE)
forest(x, alim=c(-2,1), cex=.8, header=TRUE)
```

```
### meta-analysis of the (log) risk ratios using the Mantel-Haenszel method
res <- rma.mh(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg,
              slab=paste(author, year, sep=", "))

### draw cumulative forest plot
x <- cumul(res, order=dat$year)
forest(x, alim=c(-2,1), cex=.8, header=TRUE)
```

forest.default *Forest Plots (Default Method)*

Description

Function to create forest plots for a given set of data.

Usage

```
## Default S3 method:
forest(x, vi, sei, ci.lb, ci.ub,
       annotate=TRUE, showweights=FALSE, header=FALSE,
       xlim, alim, olim, ylim, top=3, at, steps=5,
       level=95, refline=0, digits=2L, width,
       xlab, slab, ilab, ilab.xpos, ilab.pos,
       order, subset, transf, atransf, targs, rows,
       efac=1, pch=15, psize, plim=c(0.5,1.5), col,
       lty, fonts, cex, cex.lab, cex.axis, annosym, ...)
```

Arguments

`x` vector of length k with the observed effect sizes or outcomes.

`vi` vector of length k with the corresponding sampling variances.

`sei` vector of length k with the corresponding standard errors (note: only one of the two, `vi` or `sei`, needs to be specified).

`ci.lb` vector of length k with the corresponding lower confidence interval bounds. Not needed if `vi` or `sei` is specified. See ‘Details’.

`ci.ub` vector of length k with the corresponding upper confidence interval bounds. Not needed if `vi` or `sei` is specified. See ‘Details’.

`annotate` logical to specify whether annotations should be added to the plot (the default is `TRUE`).

`showweights` logical to specify whether the annotations should also include inverse variance weights (the default is `FALSE`).

`header` logical to specify whether column headings should be added to the plot (the default is `FALSE`). Can also be a character vector to specify the left and right headings.

`xlim` horizontal limits of the plot region. If unspecified, the function tries to set the horizontal plot limits to some sensible values.

`alim` the actual x-axis limits. If unspecified, the function tries to set the x-axis limits to some sensible values.

`olim` optional argument to specify observation/outcome limits. If unspecified, no limits are used.

`ylim` the y-axis limits of the plot. If unspecified, the function tries to set the y-axis limits to some sensible values.

`top` the amount of space to leave empty at the top of the plot (e.g., for adding headers) (the default is 3 rows).

`at` position of the x-axis tick marks and corresponding labels. If unspecified, the function tries to set the tick mark positions/labels to some sensible values.

`steps` the number of tick marks for the x-axis (the default is 5). Ignored when the positions are specified via the `at` argument.

`level` numeric value between 0 and 100 to specify the confidence interval level (the default is 95).

`refline` numeric value to specify the location of the vertical ‘reference’ line (the default is 0). The line can be suppressed by setting this argument to `NA`.

`digits` integer to specify the number of decimal places to which the tick mark labels of the x-axis and the annotations should be rounded (the default is `2L`). Can also be a vector of two integers, the first to specify the number of decimal places for the annotations, the second for the x-axis labels. When specifying an integer (e.g., `2L`), trailing zeros after the decimal mark are dropped for the x-axis labels. When specifying a numeric value (e.g., `2`), trailing zeros are retained.

`width` optional integer to manually adjust the width of the columns for the annotations.

`xlab` title for the x-axis. If unspecified, the function tries to set an appropriate axis title.

`slab` optional vector with labels for the k studies. If unspecified, simple labels are created within the function. To suppress labels, set this argument to `NA`.

`ilab` optional vector, matrix, or data frame providing additional information about the studies that should be added to the plot.

`ilab.xpos` numeric vector to specify the x-axis position(s) of the variable(s) given via `ilab` (must be specified if `ilab` is specified).

`ilab.pos` integer(s) (either 1, 2, 3, or 4) to specify the alignment of the vector(s) given via `ilab` (2 means right, 4 mean left aligned). If unspecified, the default is to center the labels.

`order` optional character string to specify how the studies should be ordered. Can also be a variable based on which the studies will be ordered. See ‘Details’.

`subset` optional (logical or numeric) vector to specify the subset of studies that should be included in the plot.

`transf` optional argument to specify a function that should be used to transform the observed outcomes and corresponding confidence interval bounds (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`atransf` optional argument to specify a function that should be used to transform the x-axis labels and annotations (e.g., `atransf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified via `transf` or `atransf`.

`rows` optional vector to specify the rows (or more generally, the horizontal positions) for plotting the outcomes. Can also be a single value to specify the row (horizontal position) of the first outcome (the remaining outcomes are then plotted below this starting row). If unspecified, the function sets this value automatically.

`efac` vertical expansion factor for confidence interval limits and arrows. The default value of 1 should usually work okay. Can also be a vector of two numbers, the first for CI limits, the second for arrows.

`pch` plotting symbol to use for the observed outcomes. By default, a filled square is used. See `points` for other options. Can also be a vector of values.

`psize` optional numeric value to specify the point sizes for the observed outcomes. If unspecified, the point sizes are a function of the precision of the estimates. Can also be a vector of values.

`plim` numeric vector of length 2 to scale the point sizes (ignored when `psize` is specified). See 'Details'.

`col` optional character string to specify the name of a color to use for plotting the observed outcomes (`"black"` is used by default if not specified). Can also be a vector of color names.

`lty` optional character string to specify the line type for the confidence intervals. If unspecified, the function sets this to `"solid"` by default.

`fonts` optional character string to specify the font to use for the study labels, annotations, and the extra information (if specified via `ilab`). If unspecified, the default font is used.

`cex` optional character and symbol expansion factor. If unspecified, the function tries to set this to a sensible value.

`cex.lab` optional expansion factor for the x-axis title. If unspecified, the function tries to set this to a sensible value.

`cex.axis` optional expansion factor for the x-axis labels. If unspecified, the function tries to set this to a sensible value.

`annosym` optional vector of length 3 to change the left bracket, separation, and right bracket symbols for the annotations.

`...` other arguments.

Details

The plot shows the observed effect sizes or outcomes with corresponding confidence intervals. To use the function, one should specify the observed outcomes (via the `x` argument) together with the corresponding sampling variances (via the `vi` argument) or with the corresponding standard errors (via the `sei` argument). Alternatively, one can specify the observed outcomes together with the corresponding confidence interval bounds (via the `ci.lb` and `ci.ub` arguments).

With the `transf` argument, the observed outcomes and corresponding confidence interval bounds can be transformed with some suitable function. For example, when plotting log odds ratios, then one could use transf=exp to obtain a forest plot showing the odds ratios. Alternatively, one can use the atransf argument to transform the x-axis labels and annotations (e.g., atransf=exp). See also transf for some other useful transformation functions in the context of a meta-analysis. The examples below illustrate the use of these arguments.

By default, the studies are ordered from top to bottom (i.e., the first study in the dataset will be placed in row k, the second study in row $k - 1$, and so on, until the last study, which is placed in the first row). The studies can be reordered with the order argument:

- order="obs": the studies are ordered by the observed outcomes,
- order="prec": the studies are ordered by their sampling variances.

Alternatively, it is also possible to set order equal to a variable based on which the studies will be ordered (see 'Examples').

By default (i.e., when psize is not specified), the size of the points is a function of the precision (i.e., inverse standard error) of the outcomes. This way, more precise estimates are visually more prominent in the plot. By making the point sizes a function of the inverse standard error of the estimates, their area is proportional to the inverse sampling variances, which corresponds to the weights they would receive in a fixed-effects model. However, the point sizes are rescaled so that the smallest point size is plim[1] and the largest point size is plim[2]. As a result, their relative sizes (i.e., areas) no longer exactly correspond to their relative weights in such a model. If exactly relative point sizes are desired, one can set plim[2] to NA, in which case the points are rescaled so that the smallest point size corresponds to plim[1] and all other points are scaled accordingly. As a result, the largest point may be very large. Alternatively, one can set plim[1] to NA, in which case the points are rescaled so that the largest point size corresponds to plim[2] and all other points are scaled accordingly. As a result, the smallest point may be very small and essentially indistinguishable from the confidence interval line. To avoid the latter, one can also set plim[3], which enforces a minimal point size.

Summary estimates can be added to the plot with the addpoly function. See the documentation for that function for examples.

Note

The function tries to set some sensible values for the optional arguments, but it may be necessary to adjust these in certain circumstances.

The function actually returns some information about the chosen defaults invisibly. Printing this information is useful as a starting point to make adjustments to the plot.

If the number of studies is quite large, the labels, annotations, and symbols may become quite small and impossible to read. Stretching the plot window vertically may then provide a more readable figure (one should call the function again after adjusting the window size, so that the label/symbol sizes can be properly adjusted). Also, the cex, cex.lab, and cex.axis arguments are then useful to adjust the symbol and text sizes.

If the horizontal plot and/or x-axis limits are set manually, then the horizontal plot limits (xlim) must be at least as wide as the x-axis limits (alim). This restriction is enforced inside the function.

If the outcome measure used for creating the plot is bounded (e.g., correlations are bounded between -1 and +1, proportions are bounded between 0 and 1), one can use the olim argument to enforce those limits (the observed outcomes and confidence intervals cannot exceed those bounds then).

The lty argument can also be a vector of two elements, the first for specifying the line type of the individual CIs ("solid" by default), the second for the line type of the horizontal line that is automatically added to the plot ("solid" by default; set to "blank" to remove it).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Lewis, S., & Clarke, M. (2001). Forest plots: Trying to see the wood and the trees. *British Medical Journal*, **322**(7300), 1479–1480. https://doi.org/10.1136/bmj.322.7300.1479

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

forest, forest.rma, addpoly

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### default forest plot of the observed log risk ratios
forest(dat$yi, dat$vi)

### forest plot of the observed risk ratios (transform outcomes)
forest(dat$yi, dat$vi, slab=paste(dat$author, dat$year, sep=", "), transf=exp,
       alim=c(0,2), steps=5, xlim=c(-2.5,4), refline=1, cex=.9, header=TRUE)

### forest plot of the observed risk ratios (transformed x-axis)
forest(dat$yi, dat$vi, slab=paste(dat$author, dat$year, sep=", "), atransf=exp,
       at=log(c(.05,.25,1,4,20)), xlim=c(-10,8), cex=.9, header=TRUE)

### forest plot of the observed risk ratios with studies ordered by the RRs
forest(dat$yi, dat$vi, slab=paste(dat$author, dat$year, sep=", "), atransf=exp,
       at=log(c(.05,.25,1,4,20)), xlim=c(-10,8), cex=.9, header=TRUE, order="obs")

### forest plot of the observed risk ratios with studies ordered by absolute latitude
forest(dat$yi, dat$vi, slab=paste(dat$author, dat$year, sep=", "), atransf=exp,
       at=log(c(.05,.25,1,4,20)), xlim=c(-10,8), cex=.9, header=TRUE, order=dat$ablat)

### see also examples for the forest.rma function
```

forest.rma *Forest Plots (Method for 'rma' Objects)*

Description

Function to create forest plots for objects of class "rma".

Usage

```
## S3 method for class 'rma'
forest(x, annotate=TRUE, addfit=TRUE, addpred=FALSE,
       showweights=FALSE, header=FALSE,
       xlim, alim, olim, ylim, top=3, at, steps=5,
       level=x$level, refline=0, digits=2L, width,
       xlab, slab, mlab, ilab, ilab.xpos, ilab.pos,
       order, transf, atransf, targs, rows,
       efac=1, pch=15, psize, plim=c(0.5,1.5), colout,
       col, border, lty, fonts, cex, cex.lab, cex.axis, annosym, ...)
```

Arguments

x	an object of class "rma".
annotate	logical to specify whether annotations should be added to the plot (the default is TRUE).
addfit	logical to specify whether the summary estimate (for models without moderators) or fitted values (for models with moderators) should be added to the plot (the default is TRUE). See 'Details'.
addpred	logical to specify whether the bounds of the prediction interval should be added to the plot (the default is FALSE). See 'Details'.
showweights	logical to specify whether the annotations should also include the weights given to the observed outcomes during the model fitting (the default is FALSE). See 'Details'.
header	logical to specify whether column headings should be added to the plot (the default is FALSE). Can also be a character vector to specify the left and right headings.
xlim	horizontal limits of the plot region. If unspecified, the function tries to set the horizontal plot limits to some sensible values.
alim	the actual x-axis limits. If unspecified, the function tries to set the x-axis limits to some sensible values.
olim	optional argument to specify observation/outcome limits. If unspecified, no limits are used.
ylim	the y-axis limits of the plot. If unspecified, the function tries to set the y-axis limits to some sensible values.

`top` the amount of space to leave empty at the top of the plot (e.g., for adding headers) (the default is 3 rows).

`at` position of the x-axis tick marks and corresponding labels. If unspecified, the function tries to set the tick mark positions/labels to some sensible values.

`steps` the number of tick marks for the x-axis (the default is 5). Ignored when the positions are specified via the `at` argument.

`level` numeric value between 0 and 100 to specify the confidence interval level (the default is to take the value from the object).

`refline` numeric value to specify the location of the vertical 'reference' line (the default is 0). The line can be suppressed by setting this argument to `NA`.

`digits` integer to specify the number of decimal places to which the tick mark labels of the x-axis and the annotations should be rounded (the default is `2L`). Can also be a vector of two integers, the first to specify the number of decimal places for the annotations, the second for the x-axis labels. When specifying an integer (e.g., `2L`), trailing zeros after the decimal mark are dropped for the x-axis labels. When specifying a numeric value (e.g., `2`), trailing zeros are retained.

`width` optional integer to manually adjust the width of the columns for the annotations.

`xlab` title for the x-axis. If unspecified, the function tries to set an appropriate axis title.

`slab` optional vector with labels for the k studies. If unspecified, the labels are either taken from the object (if study labels were specified) or simple labels are created within the function. To suppress labels, set this argument to `NA`.

`mlab` optional character string giving a label to the summary estimate from a fixed- or random-effects model. If unspecified, the label is created within the function.

`ilab` optional vector, matrix, or data frame providing additional information about the studies that should be added to the plot.

`ilab.xpos` numeric vector to specify the x-axis position(s) of the variable(s) given via `ilab` (must be specified if `ilab` is specified).

`ilab.pos` integer(s) (either 1, 2, 3, or 4) to specify the alignment of the vector(s) given via `ilab` (2 means right, 4 mean left aligned). If unspecified, the default is to center the labels.

`order` optional character string to specify how the studies should be ordered. Can also be a variable based on which the studies will be ordered. See 'Details'.

`transf` optional argument to specify a function that should be used to transform the observed outcomes, summary estimates, fitted values, and confidence interval bounds (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`atransf` optional argument to specify a function that should be used to transform the x-axis labels and annotations (e.g., `atransf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified via `transf` or `atransf`.

`rows` optional vector to specify the rows (or more generally, the horizontal positions) for plotting the outcomes. Can also be a single value to specify the row (horizontal position) of the first outcome (the remaining outcomes are then plotted below this starting row). If unspecified, the function sets this value automatically.

`efac` vertical expansion factor for confidence interval limits, arrows, and the symbol used to denote summary estimates. The default value of 1 should usually work okay. Can also be a vector of two numbers, the first for CI limits and arrows, the second for summary estimates. Can also be a vector of three numbers, the first for CI limits, the second for arrows, the third for summary estimates.

`pch` plotting symbol to use for the observed outcomes. By default, a filled square is used. See `points` for other options. Can also be a vector of values.

`psize` optional numeric value to specify the point sizes for the observed outcomes. If unspecified, the point sizes are a function of the model weights. Can also be a vector of values.

`plim` numeric vector of length 2 to scale the point sizes (ignored when `psize` is specified). See ‘Details’.

`colout` optional character string to specify the name of a color to use for plotting the observed outcomes (`"black"` is used by default if not specified). Can also be a vector of color names.

`col` optional character string to specify the name of a color to use for the summary polygon or fitted values. If unspecified, the function sets a default color.

`border` optional character string to specify the name of a color to use for the border of the summary polygon or fitted values. If unspecified, the function sets a default color.

`lty` optional character string to specify the line type for the confidence intervals. If unspecified, the function sets this to `"solid"` by default.

`fonts` optional character string to specify the font to use for the study labels, annotations, and the extra information (if specified via `ilab`). If unspecified, the default font is used.

`cex` optional character and symbol expansion factor. If unspecified, the function tries to set this to a sensible value.

`cex.lab` optional expansion factor for the x-axis title. If unspecified, the function tries to set this to a sensible value.

`cex.axis` optional expansion factor for the x-axis labels. If unspecified, the function tries to set this to a sensible value.

`annosym` optional vector of length 3 to change the left bracket, separation, and right bracket symbols for the annotations.

`...` other arguments.

Details

The plot shows the observed effect sizes or outcomes with corresponding confidence intervals.

For fixed- and random-effects models (i.e., for models without moderators), a four-sided polygon, sometimes called a summary ‘diamond’, is added to the bottom of the forest plot, showing the summary estimate based on the model (with the center of the polygon corresponding to the estimate and the left/right edges indicating the confidence interval limits). The `col` and `border` arguments can be used to adjust the (border) color of the polygon. Drawing of the polgyon can be suppressed by setting `addfit=FALSE`.

For random-effects models and if `addpred=TRUE`, a dotted line is added to the summary polygon which indicates the (approximate) bounds of the prediction interval (the interval indicates where `level` % of the true outcomes are expected to fall) (Riley et al., 2011). For random-effects models of class `"rma.mv"` (see `rma.mv`) with multiple τ^2 values, the `addpred` argument can be used to specify for which level of the inner factor the prediction interval should be provided (since the intervals differ depending on the τ^2 value). If the model should also contain multiple γ^2 values, the `addpred` argument should then be of length 2 to specify the levels of both inner factors. See also `predict.rma`, which is used to compute these interval bounds.

For models involving moderators, the fitted value for each study is added as a polygon to the plot. By default, the width of the polygons corresponds to the confidence interval limits for the fitted values. By setting `addpred=TRUE`, the width reflects the prediction interval limits. Again, the `col` and `border` arguments can be used to adjust the (border) color of the polygons. These polygons can be suppressed by setting `addfit=FALSE`.

With the `transf` argument, the observed outcomes, summary estimate, fitted values, confidence interval bounds, and prediction interval bounds can be transformed with some suitable function. For example, when plotting log odds ratios, one could use `transf=exp` to obtain a forest plot showing the odds ratios. Alternatively, one can use the `atransf` argument to transform the x-axis labels and annotations (e.g., `atransf=exp`). See also transf for some other useful transformation functions in the context of a meta-analysis. The examples below illustrate the use of these arguments.

By default, the studies are ordered from top to bottom (i.e., the first study in the dataset will be placed in row k, the second study in row $k-1$, and so on, until the last study, which is placed in the first row). The studies can be reordered with the `order` argument:

- `order="obs"`: the studies are ordered by the observed outcomes,
- `order="fit"`: the studies are ordered by the fitted values,
- `order="prec"`: the studies are ordered by their sampling variances,
- `order="resid"`: the studies are ordered by the size of their residuals,
- `order="rstandard"`: the studies are ordered by the size of their standardized residuals,
- `order="abs.resid"`: the studies are ordered by the size of their absolute residuals,
- `order="abs.rstandard"`: the studies are ordered by the size of their absolute standardized residuals.

Alternatively, it is also possible to set `order` equal to a variable based on which the studies will be ordered (see ‘Examples’).

Additional summary estimates can be added to the plot with the `addpoly` function. See the documentation for that function for examples.

When `showweights=TRUE`, the annotations will include information about the weights given to the observed outcomes during the model fitting. For simple models (such as those fitted with the `rma.uni` function), these weights correspond to the ‘inverse-variance weights’ (but are given in percent). For models fitted with the `rma.mv` function, the weights are based on the diagonal of the weight matrix. Note that the weighting structure is typically more complex in such models (i.e., the weight matrix is usually not just a diagonal matrix) and the weights shown therefore do not reflect this complexity. See `weights.rma` for more details.

By default (i.e., when `psize` is not specified), the size of the points is a function of the square root of the model weights. This way, their area is proportional to the the weights. However, the point sizes are rescaled so that the smallest point size is `plim[1]` and the largest point size is `plim[2]`. As a result, their relative sizes (i.e., areas) no longer exactly correspond to their relative weights. If exactly relative point sizes are desired, one can set `plim[2]` to `NA`, in which case the points are rescaled so that the smallest point size corresponds to `plim[1]` and all other points are scaled accordingly. As a result, the largest point may be very large. Alternatively, one can set `plim[1]` to `NA`, in which case the points are rescaled so that the largest point size corresponds to `plim[2]` and all other points are scaled accordingly. As a result, the smallest point may be very small and essentially indistinguishable from the confidence interval line. To avoid the latter, one can also set `plim[3]`, which enforces a minimal point size.

Note

The function tries to set some sensible values for the optional arguments, but it may be necessary to adjust these in certain circumstances.

The function actually returns some information about the chosen defaults invisibly. Printing this information is useful as a starting point to make adjustments to the plot (see 'Examples').

If the number of studies is quite large, the labels, annotations, and symbols may become quite small and impossible to read. Stretching the plot window vertically may then provide a more readable figure (one should call the function again after adjusting the window size, so that the label/symbol sizes can be properly adjusted). Also, the `cex`, `cex.lab`, and `cex.axis` arguments are then useful to adjust the symbol and text sizes.

If the horizontal plot and/or x-axis limits are set manually, then the horizontal plot limits (`xlim`) must be at least as wide as the x-axis limits (`alim`). This restriction is enforced inside the function.

If the outcome measure used for creating the plot is bounded (e.g., correlations are bounded between -1 and +1, proportions are bounded between 0 and 1), one can use the `olim` argument to enforce those limits (the observed outcomes and confidence/prediction intervals cannot exceed those bounds then).

The models without moderators, the `col` argument can also be a vector of two elements, the first for specifying the color of the summary polygon, the second for specifying the color of the line for the prediction interval.

The `lty` argument can also be a vector of up to three elements, the first for specifying the line type of the individual CIs (`"solid"` by default), the second for the line type of the prediction interval (`"dotted"` by default), the third for the line type of the horizontal lines that are automatically added to the plot (`"solid"` by default; set to `"blank"` to remove them).

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Lewis, S., & Clarke, M. (2001). Forest plots: Trying to see the wood and the trees. *British Medical Journal*, **322**(7300), 1479–1480. https://doi.org/10.1136/bmj.322.7300.1479

Riley, R. D., Higgins, J. P. T., & Deeks, J. J. (2011). Interpretation of random effects meta-analyses. *British Medical Journal*, **342**, d549. https://doi.org/10.1136/bmj.d549

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

forest, forest.default, addpoly

Examples

```
### meta-analysis of the log risk ratios using a random-effects model
res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg,
           slab=paste(author, year, sep=", "))

### default forest plot of the log risk ratios and summary estimate
forest(res, header=TRUE)

### summary estimate in row -1; studies in rows k=13 through 1; horizontal
### lines in rows 0 and k+1; two extra lines of space at the top for headings,
### and other annotations; headings (if requested) in line k+2
op <- par(xpd=TRUE)
text(x=-8.4, y=-1:16, -1:16, pos=4, cex=.6)
par(op)

### can also inspect defaults chosen
defaults <- forest(res)
defaults

### several forest plots illustrating the use of various arguments
forest(res, cex=.8)
forest(res, cex=.8, addpred=TRUE)
forest(res, cex=.8, alim=c(-3,3))
forest(res, cex=.8, order="prec", alim=c(-3,3))
forest(res, cex=.8, order=dat.bcg$ablat, addpred=TRUE)

### adjust xlim values to see how that changes the plot
forest(res)
par("usr")[1:2] ### this shows what xlim values were chosen by default
forest(res, xlim=c(-16,14))
forest(res, xlim=c(-18,10))
forest(res, xlim=c(-10,10))

### illustrate transf argument
forest(res, transf=exp, at=c(0,1,2,4,6), xlim=c(-8,12), cex=.8, refline=1, header=TRUE)

### illustrate atransf argument
forest(res, atransf=exp, at=log(c(.05,.25,1,4,20)), xlim=c(-8,7), cex=.8, header=TRUE)

### showweights argument
forest(res, atransf=exp, at=log(c(.05,.25,1,4,20)), xlim=c(-8,8),
       order="prec", showweights=TRUE, cex=.8)

### forest plot with extra annotations
### note: may need to widen plotting device to avoid overlapping text
forest(res, atransf=exp, at=log(c(.05, .25, 1, 4)), xlim=c(-16,6),
       ilab=cbind(dat.bcg$tpos, dat.bcg$tneg, dat.bcg$cpos, dat.bcg$cneg),
       ilab.xpos=c(-9.5,-8,-6,-4.5), cex=.75, header="Author(s) and Year")
```

```
op <- par(cex=.75, font=2)
text(c(-9.5,-8,-6,-4.5), 15, c("TB+", "TB-", "TB+", "TB-"))
text(c(-8.75,-5.25),     16, c("Vaccinated", "Control"))
par(op)

### mixed-effects model with absolute latitude in the model
res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, mods = ~ ablat,
           data=dat.bcg, slab=paste(author, year, sep=", "))

### forest plot with observed and fitted values
forest(res, xlim=c(-9,5), order="fit", cex=.8, ilab=dat.bcg$ablat,
       ilab.xpos=-4, atransf=exp, at=log(c(.05,.25,1,4)),
       header="Author(s) and Year")
text(-4, 15, "Latitude", cex=.8, font=2)

### meta-analysis of the log risk ratios using a random-effects model
res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg,
           slab=paste(author, year, sep=", "))

### for more complicated plots, the ylim and rows arguments may be useful
forest(res)
forest(res, ylim=c(-1.5, 16)) ### the default
forest(res, ylim=c(-1.5, 20)) ### extra space in plot
forest(res, ylim=c(-1.5, 20), rows=c(17:15, 12:6, 3:1)) ### set positions

### forest plot with subgrouping of studies
### note: may need to widen plotting device to avoid overlapping text
forest(res, xlim=c(-16, 4.6), at=log(c(.05, .25, 1, 4)), atransf=exp,
       ilab=cbind(dat.bcg$tpos, dat.bcg$tneg, dat.bcg$cpos, dat.bcg$cneg),
       ilab.xpos=c(-9.5,-8,-6,-4.5), cex=.75, ylim=c(-1, 21),
       order=dat.bcg$alloc, rows=c(1:2,5:11,14:17),
       header="Author(s) and Year")
op <- par(cex=0.75, font=2)
text(c(-9.5,-8,-6,-4.5), 20, c("TB+", "TB-", "TB+", "TB-"))
text(c(-8.75,-5.25),     21, c("Vaccinated", "Control"))
op <- par(font=4)
text(-16, c(18,12,3), c("Systematic Allocation", "Random Allocation",
                        "Alternate Allocation"), pos=4)
par(op)

### see also the addpoly.rma function for an example where summaries
### for the three subgroups are added to such a forest plot

### illustrate use of olim argument with a meta-analysis of raw correlation
### coefficients (data from Pritz, 1997); without olim=c(0,1), some of the
### CIs would have upper bounds larger than 1
dat <- escalc(measure="PR", xi=xi, ni=ni, data=dat.pritz1997)
res <- rma(yi, vi, data=dat, slab=paste0(study, ") ", authors))
forest(res, xlim=c(-0.8,1.6), alim=c(0,1), psize=1, refline=coef(res), olim=c(0,1), header=TRUE)

### an example of a forest plot where the data have a multilevel structure and
### we want to reflect this by grouping together estimates from the same cluster
dat <- dat.konstantopoulos2011
```

```
res <- rma.mv(yi, vi, random = ~ 1 | district/school, data=dat,
              slab=paste0("District ", district, ", School: ", school))
dd <- c(0,diff(dat$district))
dd[dd > 0] <- 1
rows <- (1:res$k) + cumsum(dd)
par(tck=-.01, mgp = c(1.6,.2,0))
forest(res, cex=0.5, header=TRUE, rows=rows, ylim=c(0.5,max(rows)+3))
abline(h = rows[c(1,diff(rows)) == 2] - 1, lty="dotted")
```

formula.rma *Model Formulae for 'rma' Objects*

Description

The function extracts model formulae for objects of class "rma".

Usage

```
## S3 method for class 'rma'
formula(x, type="mods", ...)
```

Arguments

x	an object of class "rma".
type	the formula which should be returned; either "mods" (default), "yi" (in case argument yi was used to specify a formula), or "scale" (only for location-scale models).
...	other arguments.

Value

The requested formula.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mv

Examples

```
### copy BCG vaccine data into 'dat'
dat <- dat.bcg

### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat,
              slab=paste(author, ", ", year, sep=""))

### mixed-effects meta-regression model
res <- rma(yi, vi, mods = ~ ablat + alloc, data=dat)
formula(res, type="mods")

### specify moderators via 'yi' argument
res <- rma(yi ~ ablat + alloc, vi, data=dat)
formula(res, type="yi")
```

`fsn` *Fail-Safe N Analysis (File Drawer Analysis)*

Description

Function to compute the fail-safe N (also called a file drawer analysis).

Usage

```
fsn(yi, vi, sei, data, type="Rosenthal", alpha=.05,
    target, weighted=FALSE, subset, digits, ...)
```

Arguments

`yi` vector with the observed effect sizes or outcomes.

`vi` vector with the corresponding sampling variances.

`sei` vector with the corresponding standard errors (note: only one of the two, `vi` or `sei`, needs to be specified).

`data` optional data frame containing the variables given to the arguments above.

`type` character string to specify the method to use for the calculation of the fail-safe N. Possible options are `"Rosenthal"` (the default), `"Orwin"`, or `"Rosenberg"`. See ‘Details’.

`alpha` target alpha level to use for the Rosenthal and Rosenberg methods (the default is .05).

`target` target average effect size or outcome to use for the Orwin method. If undefined, then the target average effect size or outcome will be equal to the observed average effect size or outcome divided by 2.

`weighted` logical to specify whether Orwin’s method should be based on unweighted (the default) or weighted averages.

`subset` optional (logical or numeric) vector to specify the subset of studies that should be used for the calculations.

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is 4.

`...` other arguments.

Details

The function can be used in conjunction with any of the usual effect sizes / outcome measures used in meta-analyses (e.g., log risk ratios, log odds ratios, risk differences, mean differences, standardized mean differences, raw correlation coefficients, correlation coefficients transformed with Fisher's r-to-z transformation, and so on). Simply specify the observed outcomes via the `yi` argument and the corresponding sampling variances via the `vi` argument (instead of specifying `vi`, one can specify the standard errors via the `sei` argument). The `escalc` function can be used to compute a wide variety of effect sizes / outcome measures (and the corresponding sampling variances) based on summary statistics.

The Rosenthal method (sometimes called a 'file drawer analysis') calculates the number of studies averaging null results that would have to be added to the given set of observed outcomes to reduce the combined significance level (p-value) to a particular alpha level (e.g., .05). The calculation is based on Stouffer's method to combine p-values and is described in Rosenthal (1979).

The Orwin method calculates the number of studies averaging null results that would have to be added to the given set of observed outcomes to reduce the (unweighted or weighted) average outcome to a target value (as specified via the `target` argument). The method is described in Orwin (1983). If `weighted=FALSE` (the default), the method does not require (or makes use) of `vi` (or `sei`), so these arguments are then not relevant for this method. If the `target` argument is not specified, then the target average outcome will be equal to the observed average outcome divided by 2 (which is quite arbitrary). One should really set `target` to a value that reflects an outcome one would consider practically irrelevant. Note that if `target` has the opposite sign as the actually observed average outcome, then its sign is automatically flipped.

The Rosenberg method calculates the number of studies averaging null results that would have to be added to the given set of observed outcomes to reduce the significance level (i.e., p-value) of the weighted average outcome (based on a fixed-effects model) to a particular alpha level (e.g., .05). The method is described in Rosenberg (2005).

If the combined/observed significance level is above the specified alpha level (for `type = "Rosenthal"` or `type = "Rosenberg"`) or if the observed average outcome is below the target average outcome (for `type = "Orwin"`), then the fail-sage N value will be 0.

Value

An object of class `"fsn"`. The object is a list containing the following components:

`type` the method used.

`fsnum` the calculated fail-safe N.

`alpha` the specified alpha level.

`pval` the p-value of the observed results. `NA` for the Orwin method.

`meanes` the average outcome of the observed results. `NA` for the Rosenthal method.

`target` the target value. `NA` for the Rosenthal and Rosenberg methods.

The results are formatted and printed with the `print.fsn` function.

Note

For the Rosenberg method, the p-value is calculated based on a standard normal distribution (instead of a t-distribution, as suggested by Rosenberg, 2005).

Author(s)

Wolfgang Viechtbauer <`wvb@metafor-project.org`> `https://www.metafor-project.org`

References

Rosenthal, R. (1979). The "file drawer problem" and tolerance for null results. *Psychological Bulletin*, **86**(3), 638–641. https://doi.org/10.1037/0033-2909.86.3.638

Orwin, R. G. (1983). A fail-safe N for effect size in meta-analysis. *Journal of Educational Statistics*, **8**(2), 157–159. https://doi.org/10.3102/10769986008002157

Rosenberg, M. S. (2005). The file-drawer problem revisited: A general weighted method for calculating fail-safe numbers in meta-analysis. *Evolution*, **59**(2), 464–468. https://doi.org/10.1111/j.0014-3820.2005.tb01004.x

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`ranktest`, `regtest`, `trimfill`

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit fixed-effects model
rma(yi, vi, data=dat, method="FE")

### fail-safe N computations
fsn(yi, vi, data=dat)
fsn(yi, data=dat, type="Orwin", target=log(0.95)) # target corresponds to a 5% risk reduction
fsn(yi, vi, data=dat, type="Orwin", weighted=TRUE, target=log(0.95))
fsn(yi, vi, data=dat, type="Rosenberg")
```

funnel *Funnel Plots*

Description

Function to create funnel plots.

Usage

```
funnel(x, ...)

## S3 method for class 'rma'
funnel(x, yaxis="sei",
       xlim, ylim, xlab, ylab,
       steps=5, at, atransf, targs, digits, level=x$level,
       addtau2=FALSE, type="rstandard",
       back="lightgray", shade="white", hlines="white",
       refline, lty=3, pch=19, pch.fill=21, col, bg,
       label=FALSE, offset=0.4, legend=FALSE, ci.res=1000, ...)

## Default S3 method:
funnel(x, vi, sei, ni, subset, yaxis="sei",
       xlim, ylim, xlab, ylab,
       steps=5, at, atransf, targs, digits, level=95,
       back="lightgray", shade="white", hlines="white",
       refline=0, lty=3, pch=19, col, bg,
       label=FALSE, offset=0.4, legend=FALSE, ci.res=1000, ...)
```

Arguments

`x` an object of class "rma" or a vector with the observed effect sizes or outcomes.

`vi` vector with the corresponding sampling variances (needed if x is a vector with the observed effect sizes or outcomes).

`sei` vector with the corresponding standard errors (note: only one of the two, vi or sei, needs to be specified).

`ni` vector with the corresponding sample sizes. Only relevant when passing a vector via x.

`subset` optional (logical or numeric) vector to specify the subset of studies that should be included in the plot. Only relevant when passing a vector via x.

`yaxis` either "sei", "vi", "seinv", "vinv", "ni", "ninv", "sqrtni", "sqrtninv", "lni", or "wi" to indicate what values should be placed on the y-axis. See 'Details'.

`xlim` x-axis limits. If unspecified, the function tries to set the x-axis limits to some sensible values.

`ylim` y-axis limits. If unspecified, the function tries to set the y-axis limits to some sensible values.

`xlab` title for the x-axis. If unspecified, the function tries to set an appropriate axis title.

`ylab` title for the y-axis. If unspecified, the function tries to set an appropriate axis title.

`steps` the number of tick marks for the y-axis (the default is 5).

`at` position of the x-axis tick marks and corresponding labels. If unspecified, the function tries to set the tick mark positions/labels to some sensible values.

`atransf` optional argument to specify a function that should be used to transform the x-axis labels (e.g., `atransf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified via `atransf`.

`digits` integer to specify the number of decimal places to which the tick mark labels of the x- and y-axis should be rounded. Can also be a vector of two integers, the first to specify the number of decimal places for the x-axis, the second for the y-axis labels (e.g., `digits=c(2,3)`). If unspecified, the function tries to set the argument to some sensible values.

`level` numeric value between 0 and 100 to specify the level of the pseudo confidence interval region (for `"rma"` objects, the default is to take the value from the object). May also be a vector of values to obtain multiple regions. See ‘Examples’.

`addtau2` logical to indicate whether the amount of heterogeneity should be accounted for when drawing the pseudo confidence interval region (the default is `FALSE`). Ignored when the model includes moderators and residuals are plotted. See ‘Details’.

`type` either `"rstandard"` (default) or `"rstudent"` to specify whether the usual or deleted residuals should be used in creating the funnel plot when the model involves moderators. See ‘Details’.

`back` color to use for the background of the plotting region (default is `"lightgray"`).

`shade` color to use for shading the pseudo confidence interval region (default is `"white"`). When `level` is a vector of values, different shading colors can be specified for each region.

`hlines` color of the horizontal reference lines (default is `"white"`).

`refline` numeric value to specify the location of the vertical ‘reference’ line and where the pseudo confidence interval should be centered. If unspecified, the reference line is drawn at the fixed- or random-effects model estimate when the model does not include moderators and at zero when moderators are included (and therefore residuals are plotted) or when directly plotting observed outcomes.

`lty` line type for the pseudo confidence interval region and the reference line. The default is to draw dotted lines (see `par` for other options). Can also be a vector to specify the two line types separately.

`pch` plotting symbol to use for the observed outcomes. By default, a filled circle is used. Can also be a vector of values. See `points` for other options.

`pch.fill` plotting symbol to use for the outcomes filled in by the trim and fill method. By default, a circle is used. Only relevant when plotting an object created by the `trimfill` function.

`col` optional character string to specify the name of a color to use for the points (`"black"` is used by default if not specified). Can also be a vector of color names.

`bg` optional character string to specify the name of a background color for open plot symbols (`"white"` is used by default if not specified). Can also be a vector of color names.

`label` argument to control the labeling of the points (the default is `FALSE`). See 'Details'.

`offset` argument to control the distance between the points and the corresponding labels.

`legend` logical to indicate whether a legend should be added to the plot (the default is `FALSE`). Can also be a keyword to indicate the position of the legend (see `legend`).

`ci.res` integer to specify the number of y-axis values at which to calculate the bounds of the pseudo confidence interval. The default is `1000`, which usually provides a sufficient resolution for the plotting.

`...` other arguments.

Details

For fixed- and random-effects models (i.e., models not involving moderators), the plot shows the observed effect sizes or outcomes on the x-axis against the corresponding standard errors (i.e., the square root of the sampling variances) on the y-axis. A vertical line indicates the estimate based on the model. A pseudo confidence interval region is drawn around this value with bounds equal to $\pm 1.96 \text{SE}$, where SE is the standard error value from the y-axis (assuming `level=95`). If `addtau2=TRUE` (only for models of class `"rma.uni"`), then the bounds of the pseudo confidence interval region are equal to $\pm 1.96\sqrt{\text{SE}^2 + \hat{\tau}^2}$, where $\hat{\tau}^2$ is the amount of heterogeneity as estimated by the model.

For models involving moderators, the plot shows the residuals on the x-axis against their corresponding standard errors. Either the usual or deleted residuals can be used for that purpose (set via the `type` argument). See `residuals.rma` for more details on the different types of residuals.

With the `atransf` argument, the labels on the x-axis can be transformed with some suitable function. For example, when plotting log odds ratios, one could use `transf=exp` to obtain a funnel plot with the values on the x-axis corresponding to the odds ratios. See also transf for some other useful transformation functions in the context of a meta-analysis.

Instead of placing the standard errors on the y-axis, several other options are available by setting the `yaxis` argument to:

- `yaxis="vi"` for the sampling variances,
- `yaxis="seinv"` for the inverse of the standard errors,
- `yaxis="vinv"` for the inverse of the sampling variances,
- `yaxis="ni"` for the sample sizes,

- `yaxis="ninv"` for the inverse of the sample sizes,
- `yaxis="sqrtni"` for the square root of the sample sizes,
- `yaxis="sqrtninv"` for the inverse square root of the sample sizes,
- `yaxis="lni"` for the log of the sample sizes,
- `yaxis="wi"` for the weights.

However, only when `yaxis="sei"` (the default) will the pseudo confidence region have the expected (upside-down) funnel shape with straight lines. Also, when placing (a function of) the sample sizes on the y-axis or the weights, then the pseudo confidence region cannot be drawn. See Sterne and Egger (2001) for more details on the choice of the y-axis.

If the object passed to the function comes from the `trimfill` function, the outcomes that are filled in by the trim and fill method are also added to the funnel plot. The symbol to use for plotting the filled in values can be specified via the `pch.fill` argument.

One can also directly pass a vector with the observed effect sizes or outcomes (via `x`) and the corresponding sampling variances (via `vi`), standard errors (via `sei`), and/or sample sizes (via `ni`) to the function. By default, the vertical reference line is then drawn at zero.

The arguments `back`, `shade`, and `hlines` can be set to `NULL` to suppress the shading and the horizontal reference line.

With the `label` argument, one can control whether points in the plot will be labeled. If `label="all"` (or `label=TRUE`), all points in the plot will be labeled. If `label="out"`, points falling outside of the pseudo confidence region will be labeled. Finally, one can also set this argument to a numeric value (between 1 and k) to specify how many of the most extreme points should be labeled (e.g., with `label=1` only the most extreme point would be labeled, while with `label=3`, the most extreme, and the second and third most extreme points would be labeled). With the `offset` argument, one can adjust the distance between the labels and the corresponding points.

Value

A data frame with components:

`x` the x-axis coordinates of the points that were plotted.

`y` the y-axis coordinates of the points that were plotted.

`slab` the study labels.

Note that the data frame is returned invisibly.

Note

Placing (a function of) the sample sizes on the y-axis (i.e., using `yaxis="ni"`, `yaxis="ninv"`, `yaxis="sqrtni"`, `yaxis="sqrtninv"`, or `yaxis="lni"`) is only possible when information about the sample sizes is actually stored within the object passed to the `funnel` function. That should automatically be the case when the observed effect sizes or outcomes were computed with the `escalc` function or when the observed effect sizes or outcomes were computed within the model fitting function. On the other hand, this will not be the case when `rma.uni` was used together with the `yi` and `vi` arguments and the `yi` and `vi` values were *not* computed with `escalc`. In that case, it is still possible to pass information about the sample sizes to the `rma.uni` function (e.g., use rma.uni(yi,vi,ni=ni,data=dat), where data frame dat includes a variable called ni with the sample sizes).

When using unweighted estimation, using yaxis="wi" will place all points on a horizontal line. When directly passing a vector with the observed effect sizes or outcomes to the function, yaxis="wi" is equivalent to yaxis="vinv", except that the weights are expressed in percent.

When specifying vectors for pch, col, and/or bg, the variables specified are assumed to be of the same length as the data passed to the funnel function or the model fitting function (when using funnel on a model object). Any subsetting and removal of studies with missing values is automatically applied to the variables specified via these arguments.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Light, R. J., & Pillemer, D. B. (1984). *Summing up: The science of reviewing research*. Cambridge, MA: Harvard University Press.

Peters, J. L., Sutton, A. J., Jones, D. R., Abrams, K. R., & Rushton, L. (2008). Contour-enhanced meta-analysis funnel plots help distinguish publication bias from other causes of asymmetry. *Journal of Clinical Epidemiology*, **61**(10), 991–996. https://doi.org/10.1016/j.jclinepi.2007.11.010

Sterne, J. A. C., & Egger, M. (2001). Funnel plots for detecting bias in meta-analysis: Guidelines on choice of axis. *Journal of Clinical Epidemiology*, **54**(10), 1046–1055. https://doi.org/10.1016/s0895-4356(01)00377-8

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mh, rma.peto, rma.glmm, rma.mv, trimfill, regtest

Examples

```
### copy BCG vaccine data into 'dat'
dat <- dat.bcg

### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat)

### fit random-effects model
res <- rma(yi, vi, data=dat, slab=paste(dat$author, dat$year, sep=", "))

### draw a standard funnel plot
funnel(res)

### show risk ratio values on x-axis (log scale)
funnel(res, atransf=exp)

### label points outside of the pseudo confidence interval region
```

```
funnel(res, atransf=exp, label=”out”)

### passing log risk ratios and sampling variances directly to the function
### note: same plot, except that reference line is centered at zero
funnel(dat$yi, dat$vi)

### can accomplish the same thing by setting refline=0
funnel(res, refline=0)

### adjust the position of the x-axis labels, number of digits, and y-axis limits
funnel(res, atransf=exp, at=log(c(.125, .25, .5, 1, 2)), digits=3L, ylim=c(0,.8))

### contour-enhanced funnel plot centered at 0 (see Peters et al., 2008)
funnel(res, level=c(90, 95, 99), shade=c(”white”, ”gray55”, ”gray75”), refline=0, legend=TRUE)

### same, but show risk ratio values on the x-axis and some further adjustments
funnel(res, level=c(90, 95, 99), shade=c(”white”, ”gray55”, ”gray75”), digits=3L, ylim=c(0,.8),
       refline=0, legend=TRUE, atransf=exp, at=log(c(.125, .25, .5, 1, 2, 4, 8)))

### illustrate the use of vectors for 'pch' and 'col'
res <- rma(yi, vi, data=dat, subset=2:10)
funnel(res, pch=ifelse(dat$yi > -1, 19, 21), col=ifelse(sqrt(dat$vi) > .3, ”red”, ”blue”))

### can add a second funnel via (undocumented) argument refline2
funnel(res, atransf=exp, at=log(c(.125, .25, .5, 1, 2, 4)), digits=3L, ylim=c(0,.8), refline2=0)

### mixed-effects model with absolute latitude in the model
res <- rma(yi, vi, mods = ~ ablat, data=dat)

### funnel plot of the residuals
funnel(res)

### simulate a large meta-analytic dataset (correlations with rho = 0.2)
### with no heterogeneity or publication bias; then try out different
### versions of the funnel plot

gencor <- function(rhoi, ni) {
   x1 <- rnorm(ni, mean=0, sd=1)
   x2 <- rnorm(ni, mean=0, sd=1)
   x3 <- rhoi*x1 + sqrt(1-rhoi^2)*x2
   cor(x1, x3)
}

set.seed(1234)
k  <- 200                               ### number of studies to simulate
ni <- round(rchisq(k, df=2) * 20 + 20)  ### simulate sample sizes (skewed distribution)
ri <- mapply(gencor, rep(0.2,k), ni)    ### simulate correlations

res <- rma(measure=”ZCOR”, ri=ri, ni=ni, method=”FE”) ### use r-to-z transformed correlations

funnel(res, yaxis=”sei”)
funnel(res, yaxis=”vi”)
funnel(res, yaxis=”seinv”)
funnel(res, yaxis="vinv")
funnel(res, yaxis="ni")
funnel(res, yaxis="ninv")
funnel(res, yaxis="sqrtni")
funnel(res, yaxis="sqrtninv")
funnel(res, yaxis="lni")
funnel(res, yaxis="wi")
```

gosh *GOSH Plots for 'rma' Objects*

Description

Function to create GOSH plots for objects of class "rma".

Usage

```
gosh(x, ...)

## S3 method for class 'rma'
gosh(x, subsets, progbar=TRUE, parallel="no", ncpus=1, cl=NULL, ...)
```

Arguments

x	an object of class "rma".
subsets	optional integer to specify the number of subsets.
progbar	logical to specify whether a progress bar should be shown (the default is TRUE).
parallel	character string to specify whether parallel processing should be used (the default is "no"). For parallel processing, set to either "snow" or "multicore". See 'Details'.
ncpus	integer to specify the number of processes to use in the parallel processing.
cl	optional cluster to use if parallel="snow". If not supplied, a cluster on the local machine is created for the duration of the call.
...	other arguments.

Details

The model specified via x must be a model fitted with either the rma.uni, rma.mh, or rma.peto function.

Olkin et al. (2012) proposed the GOSH (graphical display of study heterogeneity) plot, which is based on examining the results of a fixed-effects model in all possible subsets of size $1, \ldots, k$ of the k studies included in a meta-analysis. In a homogeneous set of studies, the model estimates obtained this way should form a roughly symmetric, contiguous, and unimodal distribution. On the other hand, when the distribution is multimodal, then this suggests the presence of heterogeneity, possibly due to outliers and/or distinct subgroupings of studies. Plotting the estimates against some measure of heterogeneity (e.g., I^2, H^2, or the Q-statistic) can also help to reveal subclusters, which are indicative of heterogeneity. The same type of plot can be produced by first fitting a fixed-effects model with either the rma.uni (using method="FE"), rma.mh, or rma.peto functions and then passing the fitted model object to the gosh function and then plotting the results.

For models fitted with the rma.uni function (which may involve moderators and/or may be random/mixed-effects models), the idea underlying this type of plot can be generalized (Viechtbauer, 2021) by examining the distribution of all model coefficients, plotting them against each other, and against some measure of (residual) heterogeneity (including the estimate of τ^2).

Note that for models without moderators, application of the method requires fitting a total of $2^k - 1$ models, which could be an excessively large number when k is large. For example, for $k = 10$, there are only 1023 possible subsets, but for $k = 20$, this number already grows to 1048575. For even larger k, it may become computationally infeasible to consider all possible subsets. Instead, we can then examine (a sufficiently large number of) random subsets.

By default, if the number of possible subsets is $\leq 10^6$, the function will consider all possible subsets and otherwise 10^6 random subsets. One can use the subsets argument to specify a different number of subsets to consider. If subsets is specified and it is actually larger than the number of possible subsets, then the function automatically only considers the possible subsets and does not use random subsets.

On machines with multiple cores, one can try to speed things up by delegating the model fitting to separate worker processes, that is, by setting parallel="snow" or parallel="multicore" and ncpus to some value larger than 1. Parallel processing makes use of the parallel package, using the makePSOCKcluster and parLapply functions when parallel="snow" or using mclapply when parallel="multicore" (the latter only works on Unix/Linux-alikes). With parallel::detectCores(), one can check on the number of available cores on the local machine.

Value

An object of class "gosh.rma". The object is a list containing the following components:

res a data frame with the results for each subset (various heterogeneity statistics and the model coefficient(s)).

incl a matrix indicating which studies were included in which subset.

... some additional elements/values.

The results can be printed with the print.gosh.rma function and plotted with the plot.gosh.rma function.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Olkin, I., Dahabreh, I. J., & Trikalinos, T. A. (2012). GOSH - a graphical display of study heterogeneity. *Research Synthesis Methods*, **3**(3), 214–223. https://doi.org/10.1002/jrsm.1053

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Viechtbauer, W. (2021). Model checking in meta-analysis. In C. H. Schmid, T. Stijnen, & I. R. White (Eds.), *Handbook of meta-analysis* (pp. 219–254). Boca Raton, FL: CRC Press. https://doi.org/10.1201/97813151194

See Also

rma.uni, rma.mh, rma.peto, influence.rma.uni,

Examples

```
### calculate log odds ratios and corresponding sampling variances
dat <- escalc(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat.egger2001)

### meta-analysis of all trials including ISIS-4 using a FE model
res <- rma(yi, vi, data=dat, method="FE")

### fit FE model to all possible subsets (65535 models)
## Not run:
sav <- gosh(res, progbar=FALSE)
sav

### create GOSH plot
### red points for subsets that include and blue points
### for subsets that exclude study 16 (the ISIS-4 trial)
plot(sav, out=16, breaks=100)
## End(Not run)
```

---

hc *Meta-Analysis based on the Method by Henmi and Copas (2010)*

Description

The function can be used to obtain an estimate of the average true outcome and corresponding confidence interval under a random-effects model using the method described by Henmi and Copas (2010).

Usage

```
hc(object, ...)

## S3 method for class 'rma.uni'
hc(object, digits, transf, targs, control, ...)
```

Arguments

`object` an object of class "rma.uni".

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.

`transf` optional argument to specify a function that should be used to transform the estimate and the corresponding interval bounds (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

targs optional arguments needed by the function specified under transf.

control list of control values for the iterative algorithm. If unspecified, default values are defined inside the function. See 'Note'.

... other arguments.

Details

The model specified via object must be a model without moderators (i.e., either a fixed- or a random-effects model and not a fixed-effects with moderators or mixed-effects model).

When using the usual method for fitting a random-effects model (i.e., weighted estimation with inverse-variance weights), the weights assigned to smaller and larger studies become more uniform as the amount of heterogeneity increases. As a consequence, the estimated average outcome could become increasingly biased under certain forms of publication bias (where smaller studies on one side of the funnel plot are missing). The method by Henmi and Copas (2010) tries to counteract this problem by providing an estimate of the average true outcome that is based on inverse-variance weights as used under a fixed-effects model (which do not take the amount of heterogeneity into consideration). The amount of heterogeneity is still estimated (with the DerSimonian-Laird estimator) and incorporated into the standard error of the estimated average outcome and the corresponding confidence interval.

Currently, there is only a method for handling objects of class "rma.uni" with the hc function. It therefore provides a method for conducting a sensitivity analysis after the model has been fitted with the rma.uni function.

Value

An object of class "hc.rma.uni". The object is a list containing the following components:

beta estimated average true outcome.

se corresponding standard error.

ci.lb lower bound of the confidence intervals for the average true outcome.

ci.ub upper bound of the confidence intervals for the average true outcome.

... some additional elements/values.

The results are formatted and printed with the print.hc.rma.uni function.

Note

The method makes use of the uniroot function. By default, the desired accuracy is set equal to .Machine$double.eps^0.25 and the maximum number of iterations to 1000. The desired accuracy (tol) and the maximum number of iterations (maxiter) can be adjusted with the control argument (i.e., control=list(tol=value,maxiter=value)).

Author(s)

Original code by Henmi and Copas (2010). Corrected for typos by Michael Dewey (<lists@dewey.myzen.co.uk>). Incorporated into the package with some small adjustments for consistency with the other functions in the package by Wolfgang Viechtbauer (<wvb@metafor-project.org>).

References

Henmi, M., & Copas, J. B. (2010). Confidence intervals for random effects meta-analysis and robustness to publication bias. *Statistics in Medicine*, **29**(29), 2969–2983. https://doi.org/10.1002/sim.4029

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni

Examples

```
### calculate log odds ratios and corresponding sampling variances
dat <- escalc(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat.lee2004)
dat

### meta-analysis based on log odds ratios
res <- rma(yi, vi, data=dat)
res

### funnel plot as in Henmi and Copas (2010)
funnel(res, yaxis="seinv", refline=0, xlim=c(-3,3), ylim=c(.5,3.5), steps=7, digits=1, back="white")

### use method by Henmi and Copas (2010) as a sensitivity analysis
hc(res)

### back-transform results to odds ratio scale
hc(res, transf=exp)
```

influence.rma.mv *Outlier and Influential Case Diagnostics for 'rma.mv' Objects*

Description

The functions can be used to compute various outlier and influential case diagnostics (some of which indicate the influence of deleting one case at a time on the model fit or the fitted/residual values) for objects of class "rma.mv".

Usage

```
## S3 method for class 'rma.mv'
cooks.distance(model, progbar=FALSE, cluster,
               reestimate=TRUE, parallel="no", ncpus=1, cl=NULL, ...)

## S3 method for class 'rma.mv'
dfbetas(model, progbar=FALSE, cluster,
        reestimate=TRUE, parallel="no", ncpus=1, cl=NULL, ...)
```

```
## S3 method for class 'rma.mv'
hatvalues(model, type="diagonal", ...)
```

Arguments

model — an object of class "rma.mv".

progbar — logical to specify whether a progress bar should be shown (the default is FALSE).

cluster — optional vector to specify a clustering variable to use for computing the Cook's distances or DFBETAS values. If not specified, these measures are computed for the individual observed effect sizes or outcomes.

reestimate — logical to specify whether variance/correlation components should be re-estimated after deletion of the ith case (the default is TRUE).

parallel — character string to specify whether parallel processing should be used (the default is "no"). For parallel processing, set to either "snow" or "multicore". See 'Details'.

ncpus — integer to specify the number of processes to use in the parallel processing.

cl — optional cluster to use if parallel="snow". If not supplied, a cluster on the local machine is created for the duration of the call.

type — character string to specify whether only the diagonal of the hat matrix ("diagonal") or the entire hat matrix ("matrix") should be returned.

... — other arguments.

Details

The term 'case' below refers to a particular row from the dataset used in the model fitting (when argument cluster is not specified) or each level of the variable specified via cluster.

Cook's distance for the ith case can be interpreted as the Mahalanobis distance between the entire set of predicted values once with the ith case included and once with the ith case excluded from the model fitting.

The DFBETAS value(s) essentially indicate(s) how many standard deviations the estimated coefficient(s) change(s) after excluding the ith case from the model fitting.

Value

The cooks.distance function returns a vector. The dfbetas function returns a data frame. The hatvalues function returns either a vector with the diagonal elements of the hat matrix or the entire hat matrix.

Note

Right now, leave-one-out diagnostics are calculated by refitting the model k times (where k is the number of cases). Depending on how large k is, it may take a few moments to finish the calculations. For complex models fitted with rma.mv, this can become computationally expensive.

On machines with multiple cores, one can usually speed things up by delegating the model fitting to separate worker processes, that is, by setting parallel="snow" or parallel="multicore" and ncpus to some value larger than 1. Parallel processing makes use of the parallel package, using the makePSOCKcluster and parLapply functions when parallel="snow" or using mclapply when parallel="multicore" (the latter only works on Unix/Linux-alikes). With parallel::detectCores(), one can check on the number of available cores on the local machine.

Alternatively (or in addition to using parallel processing), one can also set reestimate=FALSE, in which case any variance/correlation components in the model are not re-estimated after deleting the ith case from the dataset. Doing so only yields an approximation to the Cook's distances and DFBETAS values that ignores the influence of the ith case on the variance/correlation components, but is considerably faster (and often yields similar results).

It may not be possible to fit the model after deletion of the ith case from the dataset. This will result in NA values for that case.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Belsley, D. A., Kuh, E., & Welsch, R. E. (1980). *Regression diagnostics*. New York: Wiley.

Cook, R. D., & Weisberg, S. (1982). *Residuals and influence in regression*. London: Chapman and Hall.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Viechtbauer, W., & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**(2), 112–125. https://doi.org/10.1002/jrsm.11

See Also

rstudent.rma.mv, weights.rma.mv

Examples

```
### copy data from Konstantopoulos (2011) into 'dat'
dat <- dat.konstantopoulos2011

### multilevel random-effects model
res <- rma.mv(yi, vi, random = ~ 1 | district/school, data=dat)
print(res, digits=3)

### Cook's distances for each observed outcome
x <- cooks.distance(res)
x
plot(x, type="o", pch=19, xlab="Observed Outcome", ylab="Cook's Distance")

### Cook's distances for each district
x <- cooks.distance(res, cluster=dat$district)
x
plot(x, type="o", pch=19, xlab="District", ylab="Cook's Distance", xaxt="n")
axis(side=1, at=seq_along(x), labels=as.numeric(names(x)))
```

```
### hat values
hatvalues(res)
```

influence.rma.uni — *Outlier and Influential Case Diagnostics for 'rma.uni' Objects*

Description

The functions can be used to compute various outlier and influential case diagnostics (some of which indicate the influence of deleting one case at a time on the model fit or the fitted/residual values) for objects of class "rma.uni". For the corresponding help file for "rma.mv" objects, see influence.rma.mv.

Usage

```
## S3 method for class 'rma.uni'
influence(model, digits, progbar=FALSE, ...)

## S3 method for class 'infl.rma.uni'
print(x, digits=x$digits, infonly=FALSE, ...)

## S3 method for class 'rma.uni'
cooks.distance(model, progbar=FALSE, ...)
## S3 method for class 'rma.uni'
dfbetas(model, progbar=FALSE, ...)
## S3 method for class 'rma.uni'
hatvalues(model, type="diagonal", ...)
```

Arguments

model	an object of class "rma.uni".
x	an object of class "infl.rma.uni" (for print).
digits	integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.
progbar	logical to specify whether a progress bar should be shown (the default is FALSE).
infonly	logical to specify whether only the influential cases should be printed (the default is FALSE).
type	character string to specify whether only the diagonal of the hat matrix ("diagonal") or the entire hat matrix ("matrix") should be returned.
...	other arguments.

Details

The term ‘case’ below refers to a particular row from the dataset used in the model fitting (which is typically synonymous with study).

The `influence` function calculates the following leave-one-out diagnostics for each case:

- externally standardized residual,
- DFFITS value,
- Cook’s distance,
- covariance ratio,
- the leave-one-out amount of (residual) heterogeneity,
- the leave-one-out test statistic of the test for (residual) heterogeneity,
- DFBETAS value(s).

The diagonal elements of the hat matrix and the weights (in %) given to the observed effect sizes or outcomes during the model fitting are also provided (except for their scaling, the hat values and weights are the same for models without moderators, but will differ when moderators are included).

For details on externally standardized residuals, see `rstudent.rma.uni`.

The DFFITS value essentially indicates how many standard deviations the predicted (average) effect or outcome for the ith case changes after excluding the ith case from the model fitting.

Cook’s distance can be interpreted as the Mahalanobis distance between the entire set of predicted values once with the ith case included and once with the ith case excluded from the model fitting.

The covariance ratio is defined as the determinant of the variance-covariance matrix of the parameter estimates based on the dataset with the ith case removed divided by the determinant of the variance-covariance matrix of the parameter estimates based on the complete dataset. A value below 1 therefore indicates that removal of the ith case yields more precise estimates of the model coefficients.

The leave-one-out amount of (residual) heterogeneity is the estimated value of τ^2 based on the dataset with the ith case removed. This is always equal to 0 for fixed-effects models.

Similarly, the leave-one-out test statistic of the test for (residual) heterogeneity is the value of the test statistic of the test for (residual) heterogeneity calculated based on the dataset with the ith case removed.

Finally, the DFBETAS value(s) essentially indicate(s) how many standard deviations the estimated coefficient(s) change(s) after excluding the ith case from the model fitting.

A case may be considered to be ‘influential’ if at least one of the following is true:

- The absolute DFFITS value is larger than $3 \times \sqrt{p/(k-p)}$, where p is the number of model coefficients and k the number of cases.
- The lower tail area of a chi-square distribution with p degrees of freedom cut off by the Cook’s distance is larger than 50%.
- The hat value is larger than $3 \times (p/k)$.
- Any DFBETAS value is larger than 1.

Cases which are considered influential with respect to any of these measures are marked with an asterisk. Note that the chosen cut-offs are (somewhat) arbitrary. Substantively informed judgment should always be used when examining the influence of each case on the results.

Value

An object of class "infl.rma.uni", which is a list containing the following components:

`inf` an element of class "list.rma" with the externally standardized residuals, DFFITS values, Cook's distances, covariance ratios, leave-one-out τ^2 estimates, leave-one-out (residual) heterogeneity test statistics, hat values, weights, and an indicator whether a case is influential or not.

`dfbs` an element of class "list.rma" with the the DFBETAS values.

`...` some additional elements/values.

The results are printed with `print.infl.rma.uni` and plotted with `plot.infl.rma.uni`.

Note

Right now, leave-one-out diagnostics are calculated by refitting the model k times. Depending on how large k is, it may take a few moments to finish the calculations. There are shortcuts for calculating at least some of these values without refitting the model each time, but these are currently not implemented (and may not exist for all of the leave-one-out diagnostics calculated by the function).

It may not be possible to fit the model after deletion of the ith case from the dataset. This will result in `NA` values for that case.

Certain relationships between the leave-one-out diagnostics and the (internally or externally) standardized residuals (Belsley, Kuh, & Welsch, 1980; Cook & Weisberg, 1982) no longer hold for the meta-analytic models. Maybe there are other relationships. These remain to be determined.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Belsley, D. A., Kuh, E., & Welsch, R. E. (1980). *Regression diagnostics*. New York: Wiley.

Cook, R. D., & Weisberg, S. (1982). *Residuals and influence in regression*. London: Chapman and Hall.

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. San Diego, CA: Academic Press.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Viechtbauer, W., & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**(2), 112–125. https://doi.org/10.1002/jrsm.11

See Also

`plot.infl.rma.uni`, `rstudent.rma.uni`, `weights.rma.uni`

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)

### compute the diagnostics
inf <- influence(res)
inf

### plot the values
plot(inf)

### compute Cook's distances, DFBETAS values, and hat values
cooks.distance(res)
dfbetas(res)
hatvalues(res)
```

labbe — *L'Abbe Plots for 'rma' Objects*

Description

Function to create L'Abbé plots for objects of class "rma".

Usage

```
labbe(x, ...)

## S3 method for class 'rma'
labbe(x, xlim, ylim, xlab, ylab,
      add=x$add, to=x$to, transf, targs,
      pch=21, psize, plim=c(0.5,3.5),
      col, bg, grid=FALSE, lty, ...)
```

Arguments

`x` an object of class "rma". See 'Details'.

`xlim` x-axis limits. If unspecified, the function tries to set the x-axis limits to some sensible values.

`ylim` y-axis limits. If unspecified, the function tries to set the y-axis limits to some sensible values.

`xlab` title for the x-axis. If unspecified, the function tries to set an appropriate axis title.

`ylab` title for the y-axis. If unspecified, the function tries to set an appropriate axis title.

`add` See ‘Details’ and the documentation of the `escalc` function for more details.

`to` See ‘Details’ and the documentation of the `escalc` function for more details.

`transf` optional argument to specify a function that should be used to transform the outcomes (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified under `transf`.

`pch` plotting symbol to use for the outcomes. By default, a filled circle is used. Can also be a vector of values. See `points` for other options.

`psize` optional numeric vector to specify the point sizes for the outcomes. If unspecified, the point sizes are a function of the precision of the outcomes. Can also be a vector of values.

`plim` numeric vector of length 2 to scale the point sizes (ignored when `psize` is specified). See ‘Details’.

`col` optional character string to specify the name of a color to use for the points (`"black"` is used by default if not specified). Can also be a vector of color names.

`bg` optional character string to specify the name of a background color for open plot symbols (`"gray"` is used by default if not specified). Can also be a vector of color names. Set to `NA` to make the plotting symbols transparent.

`grid` logical to specify whether a grid should be added to the plot (can also be a color name).

`lty` optional character vector to specify the line type for the diagonal reference line of no effect and the line that indicates the estimated effect based on the fitted model. If unspecified, the function sets this to `c("solid","dashed")` by default (use `"blank"` to suppress a line).

`...` other arguments.

Details

The model specified via `x` must be a model without moderators (i.e., either a fixed- or a random-effects model) fitted with either the `rma.uni`, `rma.mh`, `rma.peto`, or `rma.glmm` functions. Moreover, the model must have been fitted with `measure` set equal to `"RD"` (for risk differences), `"RR"` (for risk ratios), `"OR"` (for odds ratios), `"AS"` (for arcsine square root transformed risk differences), `"IRR"` (for incidence rate ratios), `"IRD"` (for incidence rate differences), or `"IRSD"` (for square root transformed incidence rate differences).

The function calculates the arm-level outcomes for the two groups (e.g., treatment and control) and plots them against each other. In particular, the function plots the raw proportions of the two groups against each other when analyzing risk differences, the log of the proportions when analyzing (log) risk ratios, the log odds when analyzing (log) odds ratios, the arcsine square root transformed proportions when analyzing arcsine square root transformed risk differences, the raw incidence rates when analyzing incidence rate differences, the log of the incidence rates when analyzing (log) incidence rate ratios, and the square root transformed incidence rates when analyzing square root transformed incidence rate differences. The `transf` argument can be used to transform these values (e.g., `transf=exp` to transform the log of the proportions back to raw proportions; see also transf).

As described under the documentation for the escalc function, zero cells can lead to problems when calculating particular outcomes. Adding a small constant to the cells of the 2 × 2 tables is a common solution to this problem. By default, the functions adopts the same method for handling zero cells as was used when fitting the model.

By default (i.e., when `psize` is not specified), the size of the points is a function of the precision (i.e., inverse standard error) of the outcomes. This way, more precise estimates are visually more prominent in the plot. By making the point sizes a function of the inverse standard error of the estimates, their area is proportional to the inverse sampling variances, which corresponds to the weights they would receive in a fixed-effects model. However, the point sizes are rescaled so that the smallest point size is `plim[1]` and the largest point size is `plim[2]`. As a result, their relative sizes (i.e., areas) no longer exactly correspond to their relative weights in such a model. If exactly relative point sizes are desired, one can set `plim[2]` to `NA`, in which case the points are rescaled so that the smallest point size corresponds to `plim[1]` and all other points are scaled accordingly. As a result, the largest point may be very large. Alternatively, one can set `plim[1]` to `NA`, in which case the points are rescaled so that the largest point size corresponds to `plim[2]` and all other points are scaled accordingly.

The solid line corresponds to identical outcomes in the two groups (i.e., the absence of a difference between the two groups). The dashed line indicates the estimated effect based on the fitted model.

Value

A data frame with components:

`x`	the x-axis coordinates of the points that were plotted.
`y`	the y-axis coordinates of the points that were plotted.
`cex`	the point sizes.
`pch`	the plotting symbols.
`col`	the point colors.
`bg`	the background colors.
`ids`	the study id numbers.
`slab`	the study labels.

Note that the data frame is returned invisibly.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

L'Abbé, K. A., Detsky, A. S., & O'Rourke, K. (1987). Meta-analysis in clinical research. *Annals of Internal Medicine*, **107**(2), 224–233. https://doi.org/10.7326/0003-4819-107-2-224

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mh, rma.peto, rma.glmm

Examples

```
### meta-analysis of the log risk ratios using a random-effects model
res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### default plot
labbe(res)

### funnel plot with risk values on the x- and y-axis and add grid
labbe(res, transf=exp, grid=TRUE)
```

---

leave1out *Leave-One-Out Diagnostics for 'rma' Objects*

Description

The functions repeatedly fit the specified model, leaving out one observation/study at a time.

Usage

```
leave1out(x, ...)

## S3 method for class 'rma.uni'
leave1out(x, digits, transf, targs, progbar=FALSE, ...)
## S3 method for class 'rma.mh'
leave1out(x, digits, transf, targs, progbar=FALSE, ...)
## S3 method for class 'rma.peto'
leave1out(x, digits, transf, targs, progbar=FALSE, ...)
```

Arguments

`x`	an object of class `"rma.mh"`, `"rma.peto"`, or `"rma.uni"`.
`digits`	integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.
`transf`	an optional argument to specify a function that should be used to transform the model coefficients and interval bounds (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.
`targs`	optional arguments needed by the function specified under `transf`.
`progbar`	logical to specify whether a progress bar should be shown (the default is `FALSE`).
`...`	other arguments.

Details

For `"rma.uni"` objects, the model specified via `x` must be a model without moderators (i.e., either a fixed- or a random-effects model).

Value

An object of class `"list.rma"`. The object is a list containing the following components:

`estimate`	estimated (average) outcomes.
`se`	corresponding standard errors.
`zval`	corresponding test statistics.
`pval`	corresponding p-values.
`ci.lb`	lower bounds of the confidence intervals.
`ci.ub`	upper bounds of the confidence intervals.
`Q`	test statistics for the test of heterogeneity.
`Qp`	corresponding p-values.
`tau2`	estimated amount of heterogeneity (only for random-effects models).
`I2`	values of I^2.
`H2`	values of H^2.

When the model was fitted with `test="t"` or `test="knha"`, then `zval` is called `tval` in the object that is returned by the function.

The object is formatted and printed with `print.list.rma`.

Note

When using the `transf` option, the transformation is applied to the estimated coefficients and the corresponding interval bounds. The standard errors are then set equal to `NA` and are omitted from the printed output.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Viechtbauer, W., & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**(2), 112–125. https://doi.org/10.1002/jrsm.11

See Also

`rma.uni`, `rma.mh`, `rma.peto`

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### random-effects model
res <- rma(yi, vi, data=dat)

### leave-one-out analysis
leave1out(res)
leave1out(res, transf=exp)

### meta-analysis of the (log) risk ratios using the Mantel-Haenszel method
res <- rma.mh(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### leave-one-out analysis
leave1out(res)
leave1out(res, transf=exp)

### meta-analysis of the (log) odds ratios using Peto's method
res <- rma.peto(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### leave-one-out analysis
leave1out(res)
leave1out(res, transf=exp)
```

llplot *Likelihood Plot of a Parameter Corresponding to an Effect Size or Outcome Measure*

Description

Function to plot the likelihood of a certain parameter corresponding to an effect size or outcome measure given the study data.

Usage

```
llplot(measure, yi, vi, sei, ai, bi, ci, di, n1i, n2i, data, subset, drop00=TRUE,
       xvals=1000, xlim, ylim, xlab, ylab, scale=TRUE,
       lty, lwd, col, level=99.99, refline=0, ...)
```

Arguments

`measure` a character string to specify for which effect size or outcome measure the likelihoods should be calculated. See ‘Details’ for possible options and how the data should then be specified.

`yi` vector with the observed effect sizes or outcomes.

`vi` vector with the corresponding sampling variances.

`sei` vector to specify the corresponding standard.

`ai` vector to specify the 2×2 table frequencies (upper left cell).

`bi` vector to specify the 2×2 table frequencies (upper right cell).

`ci` vector to specify the 2×2 table frequencies (lower left cell).

`di` vector to specify the 2×2 table frequencies (lower right cell).

`n1i` vector to specify the group sizes or row totals (first group/row).

`n2i` vector to specify the group sizes or row totals (second group/row).

`data` optional data frame containing the variables given to the arguments above.

`subset` optional (logical or numeric) vector to specify the subset of studies that should be included in the plot.

`drop00` logical to specify whether studies with no cases (or only cases) in both groups should be dropped. See 'Details'.

`xvals` integer to specify for how many distinct values the likelihood should be evaluated.

`xlim` x-axis limits. If unspecified, the function tries to set the x-axis limits to some sensible values.

`ylim` y-axis limits. If unspecified, the function tries to set the y-axis limits to some sensible values.

`xlab` title for the x-axis. If unspecified, the function tries to set an appropriate axis title.

`ylab` title for the y-axis. If unspecified, the function tries to set an appropriate axis title.

`scale` logical to specify whether the likelihood values should be scaled, so that the total area under each curve is (approximately) equal to 1.

`lty` the line types (either a single value or a vector of length k). If unspecified, the function sets the line types according to some characteristics of the likelihood function. See 'Details'.

`lwd` the line widths (either a single value or a vector of length k). If unspecified, the function sets the widths according to the sampling variances (so that the line is thicker for more precise studies and vice-versa).

`col` the line colors (either a single value or a vector of length k). If unspecified, the function uses various shades of gray according to the sampling variances (so that darker shades are used for more precise studies and vice-versa).

`level` numeric value between 0 and 100 to specify the plotting limits for each likelihood line in terms of the confidence interval (the default is 99.99).

`refline` numeric value to specify the location of the vertical 'reference' line (the default is 0). The line can be suppressed by setting this argument to `NA`.

`...` other arguments.

Details

At the moment, the function only accepts `measure="GEN"` or `measure="OR"`.

For `measure="GEN"`, one must specify arguments `yi` for the observed effect sizes or outcomes and `vi` for the corresponding sampling variances (instead of specifying `vi`, one can specify the standard errors via the sei argument). The function then plots the likelihood of the true effect size or outcome based on a normal sampling distribution with observed outcome as given by yi and variance as given by vi for each study.

For measure="OR", one must specify arguments ai, bi, ci, and di, which denote the cell frequencies of the 2×2 tables. Alternatively, one can specify ai, ci, n1i, and n2i. See escalc function for more details. The function then plots the likelihood of the true log odds ratio based on the non-central hypergeometric distribution for each 2×2 table. Since studies with no cases (or only cases) in both groups have a flat likelihood and are not informative about the odds ratio, they are dropped by default (i.e., drop00=TRUE) and are hence not drawn (if drop00=FALSE, these likelihood are indicated by dotted lines). For studies that have a single zero count, the MLE of the odds ratio is infinite and these likelihoods are indicated by dashed lines.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

van Houwelingen, H. C., Zwinderman, K. H., & Stijnen, T. (1993). A bivariate approach to meta-analysis. *Statistics in Medicine*, **12**(24), 2273–2284. https://doi.org/10.1002/sim.4780122405

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.glmm

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### draw likelihoods
llplot(measure="GEN", yi=yi, vi=vi, data=dat, lwd=1, refline=NA, xlim=c(-3,2))

### create plot (Figure 2 in van Houwelingen, Zwinderman, & Stijnen, 1993)
llplot(measure="OR", ai=b.xci, n1i=nci, ci=b.xti, n2i=nti, data=dat.collins1985a,
       lwd=1, refline=NA, xlim=c(-4,4), drop00=FALSE)
```

---

matreg *Fit Regression Models based on Correlation and Covariance Matrices*

Description

Function to fit regression models based on correlation and covariance matrices.

Usage

```
matreg(y, x, R, n, V, cov=FALSE, means, ztor=FALSE, nearPD=FALSE, level=95, digits)
```

Arguments

`y` index of the outcome variable.

`x` indices of the predictor variable(s).

`R` correlation or covariance matrix (or only the lower triangular part including the diagonal).

`n` sample size based on which the elements in the correlation/covariance matrix were computed.

`V` variance-covariance matrix of the lower triangular elements of the correlation matrix. Either `V` or `n` should be specified, not both. See 'Details'.

`cov` logical to specify whether `R` is a covariance matrix (the default is `FALSE`).

`means` optional vector to specify the means of the variables (only relevant when `cov=TRUE`).

`ztor` logical to specify whether `R` is a matrix of r-to-z transformed correlations and it should be back-transformed to raw correlations (the default is `FALSE`). See 'Details'.

`nearPD` logical to specify whether the `nearPD` function from the Matrix package should be used when the $R_{x,x}$ matrix cannot be inverted. See 'Note'.

`level` numeric value between 0 and 100 to specify the confidence interval level (the default is 95).

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is 4.

Details

Let R be a $p \times p$ correlation or covariance matrix. Let y denote the row/column of the outcome variable and x the row(s)/column(s) of the predictor variable(s) in this matrix. Let $R_{x,x}$ and $R_{x,y}$ denote the corresponding submatrices of R. Then

$$b = R_{x,x}^{-1} R_{x,y}$$

yields the standardized or raw regression coefficients (depending on whether R is a correlation or covariance matrix) when regressing the outcome variable on the predictor variable(s).

The R matrix may be computed based on a single sample of n subjects. In this case, one should specify the sample size via argument `n`. The variance-covariance matrix of the standardized regression coefficients is then given by $\text{Var}[b] = \text{MSE} \times R_{x,x}^{-1}$, where $\text{MSE} = (1 - b' R_{x,y})/(n - m)$, where m denotes the number of predictor variables. The standard errors of the regression coefficients are then given by the square root of the diagonal elements of $\text{Var}[b]$. Test statistics (in this case, t-statistics) and the corresponding p-values can then be computed as in a regular regression analysis. When R is a covariance matrix, one should set `cov=TRUE` and specify the means of the p variables via argument `means` to obtain raw regression coefficients including the intercept and corresponding standard errors.

Alternatively, R may be the result of a meta-analysis of correlation coefficients. In this case, the elements in R are pooled correlation coefficients and the variance-covariance matrix of these pooled coefficients should be specified via argument `V`. The order of elements in `V` should correspond to the order of elements in the lower triangular part of R column-wise. For example, if R is a 4×4 matrix, then the elements are $r_{2,1}$, $r_{3,1}$, $r_{4,1}$, $r_{3,2}$, $r_{4,2}$, and $r_{4,3}$ and hence `V` should be a 6×6 variance-covariance matrix of these elements. The variance-covariance matrix of the standardized regression coefficients (i.e., $\text{Var}[b]$) is then computed as a function of `V` as described in Becker (1992) using the multivariate delta method. The standard errors of the standardized regression coefficients are then given by the square root of the diagonal elements of $\text{Var}[b]$. Test statistics (in this case, z-statistics) and the corresponding p-values can then be computed in the usual manner.

In case R is the result of a meta-analysis of Fisher r-to-z transformed correlation coefficients (and hence `V` is then the corresponding variance-covariance matrix of these pooled transformed coefficients), one should set argument `ztor=TRUE`, so that the appropriate back-transformation is then applied to `R` (and `V`) within the function.

Value

An object of class `"matreg"`. The object is a list containing the following components:

`tab` a data frame with the estimated model coefficients, standard errors, test statistics, degrees of freedom (only for t-tests), p-values, and lower/upper confidence interval bounds.

`vb` the variance-covariance matrix of the estimated model coefficients.

`...` some additional elements/values.

The results are formatted and printed with the `print.matreg` function.

Note

Only the lower triangular part of `R` (and `V` if it is specified) is used in the computations.

If $R_{x,x}$ is not invertible, an error will be issued. In this case, one can set argument `nearPD=TRUE`, in which case the `nearPD` function from the Matrix package will be used to find the nearest positive semi-definite matrix, which should be invertible. The results should be treated with caution when this is done.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Becker, B. J. (1992). Using results from replicated studies to estimate linear models. *Journal of Educational Statistics*, **17**(4), 341–362. https://doi.org/10.3102/10769986017004341

Becker, B. J. (1995). Corrections to "Using results from replicated studies to estimate linear models". *Journal of Educational and Behavioral Statistics*, **20**(1), 100–102. https://doi.org/10.3102/10769986020001100

See Also

`rcalc`, `rma.mv`

Examples

```
### copy data into 'dat'
dat <- dat.craft2003

### construct dataset and var-cov matrix of the correlations
tmp <- rcalc(ri ~ var1 + var2 | study, ni=ni, data=dat)
V <- tmp$V
dat <- tmp$dat

### turn var1.var2 into a factor with the desired order of levels
dat$var1.var2 <- factor(dat$var1.var2,
  levels=c("acog.perf", "asom.perf", "conf.perf", "acog.asom", "acog.conf", "asom.conf"))

### multivariate random-effects model
res <- rma.mv(yi, V, mods = ~ var1.var2 - 1, random = ~ var1.var2 | study, struct="UN", data=dat)
res

### restructure estimated mean correlations into a 4x4 matrix
R <- matrix(NA, nrow=4, ncol=4)
R[lower.tri(R)] <- coef(res)
rownames(R) <- colnames(R) <- c("perf", "acog", "asom", "conf")
round(R, digits=3)

### check that order in vcov(res) corresponds to order in R
round(vcov(res), digits=4)

### fit regression model with 'perf' as outcome and 'acog', 'asom', and 'conf' as predictors
fit <- matreg(1, 2:4, R=R, V=vcov(res))
fit

## Not run:
### repeat the above but with r-to-z transformed correlations
dat <- dat.craft2003
tmp <- rcalc(ri ~ var1 + var2 | study, ni=ni, data=dat, rtoz=TRUE)
V <- tmp$V
dat <- tmp$dat
dat$var1.var2 <- factor(dat$var1.var2,
  levels=c("acog.perf", "asom.perf", "conf.perf", "acog.asom", "acog.conf", "asom.conf"))
res <- rma.mv(yi, V, mods = ~ var1.var2 - 1, random = ~ var1.var2 | study, struct="UN", data=dat)
R <- matrix(NA, nrow=4, ncol=4)
R[lower.tri(R)] <- coef(res)
rownames(R) <- colnames(R) <- c("perf", "acog", "asom", "conf")
fit <- matreg(1, 2:4, R=R, V=vcov(res), ztor=TRUE)
fit
## End(Not run)
```

---

metafor.news *Read News File of the Metafor Package*

Description

Read news file of the **metafor-package**.

Usage

```
metafor.news()
```

Details

The function is just a wrapper for `news(package="metafor")` which parses and displays the ‘`NEWS`’ file of the package.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Examples

```
## Not run:
metafor.news()
## End(Not run)
```

`model.matrix.rma` *Model Matrix for ’rma’ Objects*

Description

The function extracts the model matrix for objects of class `"rma"`.

Usage

```
## S3 method for class 'rma'
model.matrix(object, ...)
```

Arguments

`object`	an object of class `"rma"`.
`...`	other arguments.

Value

The model matrix.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

fitted.rma

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)

### extract the model matrix
model.matrix(res)
```

---

permutest *Permutation Tests for 'rma.uni' Objects*

Description

The function carries out permutation tests for objects of class "rma.uni".

Usage

```
permutest(x, ...)

## S3 method for class 'rma.uni'
permutest(x, exact=FALSE, iter=1000, permci=FALSE,
          progbar=TRUE, retpermdist=FALSE, digits, control, ...)
```

Arguments

x	an object of class "rma.uni".
exact	logical to specify whether an exact permutation test should be carried out or not (the default is FALSE). See 'Details'.
iter	integer to specify the number of iterations for the permutation test when not doing an exact test (the default is 1000 iterations).
permci	logical to specify whether permutation-based CIs should also be calculated (the default is FALSE). Can also be a vector of indices to specify for which coefficients a permutation-based CI should be obtained.

`progbar` logical to specify whether a progress bar should be shown (the default is `TRUE`).

`retpermdist` logical to specify whether the permutation distributions of the test statistics should be returned (the default is `FALSE`).

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.

`control` list of control values for numerical comparisons (`comptol`) and for `uniroot` (i.e., `tol` and `maxiter`). The latter is only relevant when `permci=TRUE`. See 'Note'.

`...` other arguments.

Details

For models without moderators, the permutation test is carried out by permuting the signs of the observed effect sizes or outcomes. The (two-sided) p-value of the permutation test is then equal to the proportion of times that the absolute value of the test statistic under the permuted data is as extreme or more extreme than under the actually observed data. See Follmann and Proschan (1999) for more details.

For models with moderators, the permutation test is carried out by permuting the rows of the model matrix (i.e., X). The (two-sided) p-value for a particular model coefficient is then equal to the proportion of times that the absolute value of the test statistic for the coefficient under the permuted data is as extreme or more extreme than under the actually observed data. Similarly, for the omnibus test, the p-value is the proportion of times that the test statistic for the omnibus test is as extreme or more extreme than the actually observed one. See Higgins and Thompson (2004) and Viechtbauer et al. (2015) for more details.

If `exact=TRUE`, the function will try to carry out an exact permutation test. An exact permutation test requires fitting the model to each possible permutation once. However, the number of possible permutations increases rapidly with the number of outcomes/studies (i.e., k). For models without moderators, there are 2^k possible permutations of the signs. Therefore, for $k = 5$, there are 32 possible permutations, for $k = 10$, there are already 1024, and for $k = 20$, there are over one million permutations of the signs.

For models with moderators, the increase in the number of possible permutations may be even more severe. The total number of possible permutations of the model matrix is $k!$. Therefore, for $k = 5$, there are 120 possible permutations, for $k = 10$, there are 3,628,800, and for $k = 20$, there are over 10^{18} permutations of the model matrix.

Therefore, going through all possible permutations may become infeasible. Instead of using an exact permutation test, one can set `exact=FALSE` (which is also the default). In that case, the function approximates the exact permutation-based p-value(s) by going through a smaller number (as specified by the `iter` argument) of *random* permutations. Therefore, running the function twice on the same data can yield (slightly) different p-values. Setting `iter` sufficiently large ensures that the results become stable. Note that if `exact=FALSE` and `iter` is actually larger than the number of iterations required for an exact permutation test, then an exact test will be carried out.

For models with moderators, the exact permutation test actually only requires fitting the model to each *unique* permutation of the model matrix. The number of unique permutations will be smaller than $k!$ when the model matrix contains recurring rows. This may be the case when only including categorical moderators (i.e., factors) in the model or when any quantitative moderators included in the model can only take on a small number of unique values. When exact=TRUE, the function therefore uses an algorithm to restrict the test to only the unique permutations of the model matrix, which may make the use of the exact test feasible even when k is large.

When using random permutations, the function ensures that the very first permutation will always correspond to the original data. This avoids p-values equal to 0.

When permci=TRUE, the function also tries to obtain permutation-based CIs of the model coefficient(s). This is done by shifting the observed effect sizes or outcomes by some amount and finding the most extreme values for this amount for which the permutation-based test would just lead to non-rejection. This is computationally expensive and may take a long time to complete. For models with moderators, one can also set permci to a vector of indices to specify for which coefficient(s) a permutation-based CI should be obtained. When the algorithm fails to determine a particular CI bound, it will be shown as NA in the output.

Value

An object of class "permutest.rma.uni". The object is a list containing the following components:

pval	p-value(s) based on the permutation test.
QMp	p-value for the omnibus test of moderators based on the permutation test.
zval.perm	values of the test statistics of the coefficients under the various permutations (only when retpermdist=TRUE).
b.perm	the model coefficients under the various permutations (only when retpermdist=TRUE).
QM.perm	the test statistic of the omnibus test of moderators under the various permutations (only when retpermdist=TRUE).
ci.lb	lower bound of the confidence intervals for the coefficients (permutation-based when permci=TRUE).
ci.ub	upper bound of the confidence intervals for the coefficients (permutation-based when permci=TRUE).
...	some additional elements/values are passed on.

The results are formatted and printed with the print.permutest.rma.uni function. One can also use coef.permutest.rma.uni to obtain the table with the model coefficients, corresponding standard errors, test statistics, p-values, and confidence interval bounds.

Note

The p-values obtained with permutation tests cannot reach conventional levels of statistical significance (i.e., $p \leq .05$) when k is very small. In particular, for models without moderators, the smallest possible (two-sided) p-value is .0625 when $k = 5$ and .03125 when $k = 6$. Therefore, the permutation test is only able to reject the null hypothesis at $\alpha = .05$ when k is at least equal to 6. For models with moderators, the smallest possible (two-sided) p-value for a particular model coefficient is .0833 when $k = 4$ and .0167 when $k = 5$ (assuming that each row in the model matrix is unique). Therefore, the permutation test is only able to reject the null hypothesis at $\alpha = .05$ when k is at least equal to 5. Consequently, permutation-based CIs can also only be obtained when k is sufficiently large.

When the number of permutations required for the exact test is so large as to be essentially indistinguishable from infinity (e.g., factorial(200)), the function will terminate with an error.

Determining whether a test statistic under the permuted data is as extreme or more extreme than under the actually observed data requires making >= or <= comparisons. To avoid problems due to the finite precision with which computers generally represent numbers, the function uses a numerical tolerance (control argument `comptol`, which is set equal to `.Machine$double.eps^0.5` by default) when making such comparisons (e.g., instead of `sqrt(3)^2 -3 >= 0`, which may evaluate to `FALSE`, we can use `sqrt(3)^2 -3 >= 0 -.Machine$double.eps^0.5`, which should evaluate to `TRUE`).

When obtaining permutation-based CIs, the function makes use of `uniroot`. By default, the desired accuracy is set equal to `.Machine$double.eps^0.25` and the maximum number of iterations to `100`. The desired accuracy and the maximum number of iterations can be adjusted with the `control` argument (i.e., `control=list(tol=value,maxiter=value)`).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Follmann, D. A., & Proschan, M. A. (1999). Valid inference in random effects meta-analysis. *Biometrics*, **55**(3), 732–737. https://doi.org/10.1111/j.0006-341x.1999.00732.x

Good, P. I. (2009). *Permutation, parametric, and bootstrap tests of hypotheses* (3rd ed.). New York: Springer.

Higgins, J. P. T., & Thompson, S. G. (2004). Controlling the risk of spurious findings from meta-regression. *Statistics in Medicine*, **23**(11), 1663–1682. https://doi.org/10.1002/sim.1752

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Viechtbauer, W., López-López, J. A., Sánchez-Meca, J., & Marín-Martínez, F. (2015). A comparison of procedures to test for moderators in mixed-effects meta-regression models. *Psychological Methods*, **20**(3), 360–374. https://doi.org/10.1037/met0000023

See Also

`rma.uni`, `print.permutest.rma.uni`

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### random-effects model
res <- rma(yi, vi, data=dat)
res

### permutation test (approximate and exact)
## Not run:
permutest(res)
permutest(res, exact=TRUE)
## End(Not run)

### mixed-effects model with two moderators (absolute latitude and publication year)
```

```
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)
res

### permutation test (approximate only; exact not feasible)
## Not run:

permres <- permutest(res, iter=10000, retpermdist=TRUE)
permres

### histogram of permutation distribution for absolute latitude
### dashed horizontal line: the observed value of the test statistic
### red curve: standard normal density
### blue curve: kernel density estimate of the permutation distribution
### note that the tail area under the permutation distribution is larger
### than under a standard normal density (hence, the larger p-value)
hist(permres$zval.perm[,2], breaks=120, freq=FALSE, xlim=c(-5,5), ylim=c(0,.4),
     main="Permutation Distribution", xlab="Value of Test Statistic", col="gray90")
abline(v=res$zval[2], lwd=2, lty="dashed")
abline(v=0, lwd=2)
curve(dnorm, from=-5, to=5, add=TRUE, lwd=2, col=rgb(1,0,0,alpha=.7))
lines(density(permres$zval.perm[,2]), lwd=2, col=rgb(0,0,1,alpha=.7))

## End(Not run)
```

plot.cumul.rma *Plot Method for 'cumul.rma' Objects*

Description

Plot method for objects of class "cumul.rma".

Usage

```
## S3 method for class 'cumul.rma'
plot(x, yaxis, xlim, ylim, xlab, ylab,
     at, transf, atransf, targs, digits, cols=c("gray80","gray10"),
     grid=TRUE, pch=19, cex=1, lwd=2, ...)
```

Arguments

`x` an object of class "cumul.rma".

`yaxis` either "tau2", "I2", or "H2" to indicate what values should be placed on the y-axis. See 'Details'.

`xlim` x-axis limits. If unspecified, the function tries to set the x-axis limits to some sensible values.

`ylim` y-axis limits. If unspecified, the function tries to set the y-axis limits to some sensible values.

`xlab` title for the x-axis. If unspecified, the function tries to set an appropriate axis title.

`ylab` title for the y-axis. If unspecified, the function tries to set an appropriate axis title.

`at` position of the x-axis tick marks and corresponding labels. If unspecified, the function tries to set the tick mark positions/labels to some sensible values.

`transf` optional argument to specify a function that should be used to transform the summary estimates (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`atransf` optional argument to specify a function that should be used to transform the x-axis labels (e.g., `atransf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified via `transf` or `atransf`.

`digits` integer to specify the number of decimal places to which the tick mark labels of the x- and y-axis should be rounded. Can also be a vector of two integers, the first to specify the number of decimal places for the x-axis, the second for the y-axis labels (e.g., `digits=c(2,3)`). If unspecified, the function tries to set the argument to some sensible values.

`cols` vector with two or more colors to use for visualizing the order of the cumulative results.

`grid` logical to specify whether a grid should be added to the plot (can also be a color name).

`pch` plotting symbol to use. By default, a filled circle is used. See `points` for other options.

`cex` symbol expansion factor.

`lwd` line width.

`...` other arguments.

Details

The function can be used to visualize the results from a cumulative meta-analysis as obtained with the `cumul` function.

The plot shows the model estimate (i.e., the estimated overall/average outcome) on the x-axis and some measure of heterogeneity on the y-axis in the cumulative order of the results in the `"cumul.rma"` object. By default, τ^2 is shown on the y-axis for a random-effects model and I^2 otherwise, but one can also use argument `yaxis` to specify the measure of heterogeneity to place on the y-axis.

The color gradient of the points/lines indicates the order of the cumulative results (by default, light gray at the beginning, dark gray at the end). A different set of colors can be chosen via the `cols` argument. See ‘Examples’.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

cumul.rma.uni

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### random-effects model
res <- rma(yi, vi, data=dat)

### cumulative meta-analysis (in the order of publication year)
sav <- cumul(res, transf=exp, order=dat$year)

### plot of model estimate and tau^2 over time
plot(sav)

### illustrate some other plot options
plot(sav, yaxis="I2", ylim=c(0,100), atransf=exp, at=log(seq(1.3, 1.6, by=.1)),
     lwd=5, cex=1.5, cols=c("green","blue","red"))
```

plot.gosh.rma *Plot Method for 'gosh.rma' Objects*

Description

Plot method for objects of class "gosh.rma".

Usage

```
## S3 method for class 'gosh.rma'
plot(x, het="I2", pch=16, cex=0.5, out, col, alpha, border,
     xlim, ylim, xhist=TRUE, yhist=TRUE, hh=0.3, breaks,
     adjust, lwd, labels, ...)
```

Arguments

x	an object of class "gosh.rma".
het	character string to specify the heterogeneity measure to plot. Either "I2", "H2", "QE", or "tau2" (the last only for random/mixed-effects models).
pch	plotting symbol to use. By default, a borderless filled circle is used. See points for other options.

`cex` symbol expansion factor.

`out` optional integer to specify the number of a study that may be a potential outlier. If specified, subsets containing the specified study are drawn in a different color than those not containing the study.

`col` optional character string to specify the name of a color to use for the points (if not provided, points are drawn in black). When `out` is used, two colors should be specified (if not provided, red is used for subsets containing the specified study and blue otherwise).

`alpha` optional alpha transparency value for the points (0 means fully transparent and 1 means opaque). If unspecified, the function tries to set this to a sensible value.

`border` optional character string to specify the name of a color to use for the borders of the histogram (if not provided, borders are drawn in white). Set to `FALSE` to omit the borders.

`xlim` x-axis limits. If unspecified, the function tries to set the x-axis limits to some sensible values.

`ylim` y-axis limits. If unspecified, the function tries to set the y-axis limits to some sensible values.

`xhist` logical to specify whether a histogram should be drawn for the x-axis (the default is `TRUE`).

`yhist` logical to specify whether a histogram should be drawn for the y-axis (the default is `TRUE`).

`hh` optional numeric value (or vector of two values) to adjust the height of the histogram(s). Must be between 0 and 1, but should not be too close to 0 or 1, as otherwise the plot cannot be drawn.

`breaks` optional argument passed on to `hist` for choosing the (number of) breakpoints of the histogram(s).

`adjust` optional argument passed on to `density` for adjusting the bandwidth of the kernel density estimate(s) (values larger than 1 result in more smoothing).

`lwd` optional numeric value to specify the line width of the estimated densities. Set to `0` to omit the line(s).

`labels` optional argument to specify the x-axis and y-axis labels (or passed on to `pairs` to specify the names of the variables in the scatter plot matrix).

`...` other arguments.

Details

For models without moderators, the function draws a scatter plot of the model estimates on the x-axis against the chosen measure of heterogeneity on the y-axis. Histograms of the respective distributions (with kernel density estimates superimposed) are shown in the margins (when `xhist=TRUE` and `yhist=TRUE`).

For models with moderators, the function draws a scatter plot matrix (with the `pairs` function) of the chosen measure of heterogeneity and each of the model coefficients. Histograms of the variables plotted are shown along the diagonal, with kernel density estimates of the distributions superimposed. Arguments `xlim`, `ylim`, `xhist`, and `yhist` are then ignored (argument `hh` can then be used to compress/stretch the height of the distributions shown along the diagonal).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Olkin, I., Dahabreh, I. J., & Trikalinos, T. A. (2012). GOSH - a graphical display of study heterogeneity. *Research Synthesis Methods*, **3**(3), 214–223. https://doi.org/10.1002/jrsm.1053

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Viechtbauer, W. (2021). Model checking in meta-analysis. In C. H. Schmid, T. Stijnen, & I. R. White (Eds.), *Handbook of meta-analysis* (pp. 219–254). Boca Raton, FL: CRC Press. https://doi.org/10.1201/97813151194(

See Also

gosh.rma

Examples

```
### calculate log odds ratios and corresponding sampling variances
dat <- escalc(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat.egger2001)

### meta-analysis of all trials including ISIS-4 using a FE model
res <- rma(yi, vi, data=dat, method="FE")

### fit FE model to all possible subsets (65535 models)
## Not run:
sav <- gosh(res, progbar=FALSE)

### create GOSH plot
### red points for subsets that include and blue points
### for subsets that exclude study 16 (the ISIS-4 trial)
plot(sav, out=16, breaks=100)
## End(Not run)
```

---

plot.infl.rma.uni *Plot Method for 'infl.rma.uni' Objects*

Description

Plot method for objects of class "infl.rma.uni".

Usage

```
## S3 method for class 'infl.rma.uni'
plot(x, plotinf=TRUE, plotdfbs=FALSE, dfbsnew=FALSE, logcov=TRUE,
     layout, slab.style=1, las=0, pch=21, bg="black",
     bg.infl="red", col.na="lightgray", ...)
```

Arguments

`x` an object of class `"infl.rma.uni"`.

`plotinf` logical to specify whether the various case diagnostics should be plotted (the default is `TRUE`). Can also be a vector of up to 8 integers to specify which plots to draw. See 'Details' for the numbers corresponding to the various plots.

`plotdfbs` logical to specify whether the DFBETAS values should be plotted (the default is `FALSE`). Can also be a vector of integers to specify for which coefficient(s) to plot the DFBETAS values.

`dfbsnew` logical to specify whether a new device should be opened for plotting the DFBETAS values (the default is `FALSE`).

`logcov` logical to specify whether the covariance ratios should be plotted on a log scale (the default is `TRUE`).

`layout` optional vector of two numbers to specify the number of rows and columns for the layout of the figure.

`slab.style` integer to indicate the style of the x-axis labels: 1 = study number, 2 = study label, 3 = abbreviated study label. Note that study labels, even when abbreviated, may be too long to fit in the margins.)

`las` integer between 0 and 3 to specify the alignment of the axis labels (see `par`). The most useful alternative to 0 is 3, so that the x-axis labels are drawn vertical to the axis.

`pch` plotting symbol to use. By default, a filled circle is used. See `points` for other options.

`bg` color to use for filling the plotting symbol (the default is `"black"`).

`bg.infl` color to use for filling the plotting symbol when the point is considered influential (the default is `"red"`).

`col.na` color to use for lines connecting two points with `NA` values in between (the default is `"lightgray"`).

`...` other arguments.

Details

When `plotinf=TRUE`, the function plots the (1) externally standardized residuals, (2) DFFITS values, (3) Cook's distances, (4) covariance ratios, (5) leave-one-out τ^2 estimates, (6) leave-one-out (residual) heterogeneity test statistics, (7) hat values, and (8) weights. If `plotdfbs=TRUE`, the DFBETAS values are also plotted either after confirming the page change (if `dfbsnew=FALSE`) or on a separate device (if `dfbsnew=TRUE`).

A case (which is typically synonymous with study) may be considered to be 'influential' if at least one of the following is true:

- The absolute DFFITS value is larger than $3 \times \sqrt{p/(k-p)}$, where p is the number of model coefficients and k the number of cases.
- The lower tail area of a chi-square distribution with p degrees of freedom cut off by the Cook's distance is larger than 50%.
- The hat value is larger than $3 \times (p/k)$.

- Any DFBETAS value is larger than 1.

Cases which are considered influential with respect to any of these measures are indicated by the color specified for the `bg.infl` argument (the default is `"red"`).

The cut-offs described above are indicated in the plot with horizontal reference lines. In addition, on the plot of the externally standardized residuals, horizontal reference lines are drawn at -1.96, 0, and 1.96. On the plot of the hat values, a horizontal reference line is drawn at p/k. Since the sum of the hat values is equal to p, the value p/k indicates equal hat values for all k cases. Finally, on the plot of weights, a horizontal reference line is drawn at $100/k$, corresponding to the value for equal weights (in $\%$) for all k cases. Note that all weights will automatically be equal to each other when using unweighted model fitting. Also, the hat values will be equal to the weights values (except for their scaling) in models without moderators.

The chosen cut-offs are (somewhat) arbitrary. Substantively informed judgment should always be used when examining the influence of each case on the results.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Viechtbauer, W., & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**(2), 112–125. https://doi.org/10.1002/jrsm.11

See Also

`influence.rma.uni`

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)

### compute the diagnostics
inf <- influence(res)

### plot the values
plot(inf)

### select which plots to show
plot(inf, plotinf=1:4)
plot(inf, plotinf=1:4, layout=c(4,1))

### plot the DFBETAS values
plot(inf, plotinf=FALSE, plotdfbs=TRUE)
```

plot.rma *Plot Method for 'rma' Objects*

Description

Plot method for objects of class "rma.uni", "rma.mh", "rma.peto", and "rma.glmm".

Usage

```
## S3 method for class 'rma.uni'
plot(x, qqplot=FALSE, ...)

## S3 method for class 'rma.mh'
plot(x, qqplot=FALSE, ...)

## S3 method for class 'rma.peto'
plot(x, qqplot=FALSE, ...)

## S3 method for class 'rma.glmm'
plot(x, qqplot=FALSE, ...)
```

Arguments

x	an object of class "rma.uni", "rma.mh", or "rma.peto". The method is not yet implemented for objects of class "rma.glmm".
qqplot	logical to specify whether a normal QQ plot should be drawn (the default is FALSE).
...	other arguments.

Details

Four plots are produced. If the model does not contain any moderators, then a forest plot, funnel plot, radial plot, and a plot of the standardized residuals is provided. If qqplot=TRUE, the last plot is replaced by a normal QQ plot of the standardized residuals.

If the model contains moderators, then a forest plot, funnel plot, plot of the standardized residuals against the fitted values, and a plot of the standardized residuals is provided. If qqplot=TRUE, the last plot is replaced by a normal QQ plot of the standardized residuals.

Note

If the number of studies is large, the forest plot may become difficult to read due to the small font size. Stretching the plotting device vertically should provide more space.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

forest, funnel, radial, qqnorm.rma.uni

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit random-effects model
res <- rma(yi, vi, data=dat)

### plot results
plot(res, qqplot=TRUE)

### fit mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)

### plot results
plot(res, qqplot=TRUE)
```

---

plot.rma.uni.selmodel *Plot Method for 'plot.rma.uni.selmodel' Objects*

Description

Plot method for objects of class "plot.rma.uni.selmodel".

Usage

```
## S3 method for class 'rma.uni.selmodel'
plot(x, xlim, ylim, n=1000, prec="max", scale=FALSE,
      ci=FALSE, reps=1000, rug=TRUE, add=FALSE,
      lty=c("solid","dotted"), lwd=c(2,1), ...)
```

Arguments

x an object of class "rma.uni.selmodel".

xlim x-axis limits. Essentially the range of p-values for which the selection function should be drawn. If unspecified, the function sets the limits automatically.

ylim y-axis limits. If unspecified, the function sets the limits automatically.

n numeric value to specify for how many p-values within the x-axis limits the function value should be computed (the default is 1000).

`prec` either a character string (with options `"max"`, `"min"`, `"mean"`, or `"median"`) or a numeric value. See ‘Details’.

`scale` logical to specify whether the function values should be rescaled to a 0 to 1 range (the default is `FALSE`).

`ci` logical to specify whether a confidence interval should be drawn around the selection function (the default is `FALSE`). Can also be a string (with options `"boot"` or `"wald"`). See ‘Details’.

`reps` numeric value to specify the number of bootstrap samples to draw for generating the confidence interval bounds (the default is 1000).

`rug` logical to specify whether the observed p-values should be added as tick marks on the x-axis (the default is `TRUE`).

`add` logical to specify whether the function should be added to an existing plot (the default is `FALSE`).

`lty` the line types for the selection function and the confidence interval bounds.

`lwd` the line widths for the selection function and the confidence interval bounds.

`...` other arguments.

Details

The function can be used to draw the estimated selection function based on objects of class `"plot.rma.uni.selmodel"`.

When the selection function incorporates a measure of precision (which, strictly speaking, is really a measure of imprecision), one can specify for which level of precision the selection function should be drawn. When `prec="max"`, then the function is drawn for the *least* precise study (maximum imprecision), when `prec="min"`, then the function is drawn for the *most* precise study (minimum imprecision), while `prec="mean"` and `prec="median"` will show the function for the mean and median level of imprecision, respectively. Alternatively, one can specify a numeric value for argument `prec` to specify the precision value (where `prec="max"` corresponds to `prec=1` and higher levels of precision to `prec` values below 1).

When `ci=TRUE` (or equivalently, `ci="boot"`), a confidence interval is drawn around the selection function. The bounds of this interval are generated using parametric bootstrapping, with argument `reps` controlling the number of bootstrap samples to draw for generating the confidence interval bounds. When both `n` and `reps` are large, constructing the confidence interval can take a few moments to complete.

For models where the selection function involves a single δ parameter, one can also set `ci="wald"`, in which case the confidence interval will be constructed based on the Wald-type CI of the δ parameter (doing so is much quicker than using parametric bootstrapping). This option is also available for step function models (even if they involve multiple δ parameters).

Author(s)

Wolfgang Viechtbauer <`wvb@metafor-project.org`> `https://www.metafor-project.org`

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

selmodel.rma.uni

Examples

```
### copy data into 'dat' and examine data
dat <- dat.hackshaw1998

### fit random-effects model using the log odds ratios
res <- rma(yi, vi, data=dat, method="ML")
res

### fit step selection model
sel1 <- selmodel(res, type="stepfun", steps=c(0.05, 0.10, 0.50, 1.00))

### plot selection function
plot(sel1, scale=TRUE)

### fit negative exponential selection model
sel2 <- selmodel(res, type="negexp")

### add selection function to the existing plot
plot(sel2, add=TRUE, col="blue")

### plot selection function with CI
plot(sel1, ci="wald")
```

predict.rma *Predicted Values for 'rma' Objects*

Description

The function computes predicted values, corresponding standard errors, confidence intervals, and prediction intervals for objects of class "rma".

Usage

```
## S3 method for class 'rma'
predict(object, newmods, intercept, tau2.levels, gamma2.levels, addx=FALSE,
        level, digits, transf, targs, vcov=FALSE, ...)

## S3 method for class 'rma.ls'
predict(object, newmods, intercept, addx=FALSE, newscale, addz=FALSE,
        level, digits, transf, targs, vcov=FALSE, ...)
```

Arguments

`object` an object of class `"rma"` or `"rma.ls"`.

`newmods` optional vector or matrix to specify the values of the moderator values for which the predicted values should be calculated. See 'Details'.

`intercept` logical to specify whether the intercept should be included when calculating the predicted values for `newmods`. If unspecified, the intercept is automatically added when the original model also included an intercept.

`tau2.levels` vector to specify the levels of the inner factor when computing prediction intervals. Only relevant for models of class `"rma.mv"` (see `rma.mv`) and when the model includes more than a single τ^2 value. See 'Details'.

`gamma2.levels` vector to specify the levels of the inner factor when computing prediction intervals. Only relevant for models of class `"rma.mv"` (see `rma.mv`) and when the model includes more than a single γ^2 value. See 'Details'.

`addx` logical to specify whether the values of the moderator variables should be added to the returned object. See 'Examples'.

`newscale` optional vector or matrix to specify the values of the scale variables for which the predicted values should be calculated. Only relevant for location-scale models (see `rma`). See 'Details'.

`addz` logical to specify whether the values of the scale variables should be added to the returned object.

`level` numeric value between 0 and 100 to specify the confidence and prediction interval level. If unspecified, the default is to take the value from the object.

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.

`transf` optional argument to specify a function that should be used to transform the predicted values and interval bounds (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified under `transf`.

`vcov` logical to specify whether the variance-covariance matrix of the predicted values should also be returned (the default is `FALSE`).

`...` other arguments.

Details

For a fixed-effects model, `predict(object)` returns the estimated (average) outcome in the set of studies included in the meta-analysis. This is the same as the estimated intercept in the fixed-effects model (i.e., $\hat{\theta}$).

For a random-effects model, `predict(object)` returns the estimated (average) outcome in the hypothetical population of studies from which the set of studies included in the meta-analysis are assumed to be a random selection. This is the same as the estimated intercept in the random-effects model (i.e., $\hat{\mu}$).

For models including one or more moderators, `predict(object)` returns the estimated (average) outcomes for values of the moderator(s) equal to those of the k studies included in the meta-analysis (i.e., the ‘fitted values’ for the k studies).

For models including p' moderator variables, new moderator values (for k_{new} hypothetical new studies) can be specified by setting `newmods` equal to a $k_{new} \times p'$ matrix with the corresponding new moderator values. If the model object included an intercept, then it should not be explicitly specified under `newmods`, as it will be added by default (unless one sets `intercept=FALSE`). Also, any factors in the original model get turned into the appropriate contrast variables within the `rma` function, so that `newmods` should actually include the values for the contrast variables. Examples are shown below.

For random/mixed-effects models, an approximate prediction interval is also calculated (Riley et al., 2011). The interval estimates where `level` % of the true effect sizes or outcomes fall in the hypothetical population of studies (and hence where the true effect or outcome of a new study from the population of studies should fall in `level` % of the cases).

For random-effects models that were fitted with the `rma.mv` function, the model may actually include multiple τ^2 values (i.e., when the `random` argument includes an ‘`~ inner | outer`’ term and `struct="HCS"`, `struct="DIAG"`, `struct="HAR"`, or `struct="UN"`). In that case, the function will provide prediction intervals for each level of the inner factor (since the prediction intervals differ depending on the τ^2 value). Alternatively, one can use the `tau2.levels` argument to specify for which level(s) the prediction interval should be provided. If the model includes a second ‘`~ inner | outer`’ term with multiple γ^2 values, prediction intervals for each combination of levels of the inner factors will be provided. Alternatively, one can use the `tau2.levels` and `gamma2.levels` arguments to specify for which level combination(s) the prediction interval should be provided.

When using the `newmods` argument for mixed-effects models that were fitted with the `rma.mv` function, if the model includes multiple τ^2 (and multiple γ^2) values, then one must use the `tau2.levels` (and `gamma2.levels`) argument to specify the levels of the inner factor(s) (i.e., a vector of length k_{new}) to obtain the appropriate prediction interval(s).

For location-scale models fitted with the `rma` function, one can use `newmods` to specify the values of the p' moderator variables included in the model and `newscale` to specify the values of the q' scale variables included in the model. Whenever `newmods` is specified, the function computes predicted effects/outcomes for the specified moderators values. To obtain the corresponding prediction intervals, one must also specify the corresponding `newscale` values. If only `newscale` is specified (and not `newmods`), the function computes the predicted log-transformed τ^2 values (when using a log link) for to the specified scale values. By setting `transf=exp`, one can then obtain the predicted τ^2 values.

Value

An object of class `"list.rma"`. The object is a list containing the following components:

`pred`	predicted value(s).
`se`	corresponding standard error(s).
`ci.lb`	lower bound of the confidence interval(s).
`ci.ub`	upper bound of the confidence interval(s).
`pi.lb`	lower bound of the prediction interval(s) (only for random/mixed-effects models).

pi.ub upper bound of the prediction interval(s) (only for random/mixed-effects models).

tau2.level the level(s) of the inner factor (only for models of class "rma.mv" with multiple τ^2 values).

gamma2.level the level(s) of the inner factor (only for models of class "rma.mv" with multiple γ^2 values).

X the moderator value(s) used to calculate the predicted values (only when addx=TRUE).

Z the scale value(s) used to calculate the predicted values (only when addz=TRUE and only for location-scale models).

... some additional elements/values.

If vcov=TRUE, then the returned object is a list with the first element equal to the one as described above and the second element equal to the variance-covariance matrix of the predicted values.

The object is formatted and printed with print.list.rma.

Note

Confidence and prediction intervals are calculated based on the critical values from a standard normal distribution (i.e., ± 1.96 for level=95). When the model was fitted with test="t" or test="knha", then a t-distribution with $k - p$ degrees of freedom is used.

For a random-effects model (where $p = 1$) fitted with the rma.uni function, note that this differs slightly from Riley et al. (2001), who suggest to use a t-distribution with $k - 2$ degrees of freedom for constructing the prediction interval. Neither a normal, nor a t-distribution with $k - 1$ or $k - 2$ degrees of freedom is correct; all of these are approximations. The computations are done in the way described above, so that the prediction interval is identical to the confidence interval when $\hat{\tau}^2 = 0$, which could be argued is the logical thing that should happen. If the prediction interval should be computed exactly as described by Riley et al. (2001), then one can use argument pi.type="riley".

The predicted values are based only on the fixed effects of the model. Best linear unbiased predictions (BLUPs) that combine the fitted values based on the fixed effects and the estimated contributions of the random effects can be obtained with blup.rma.uni (currently only for objects of class "rma.uni").

When using the transf option, the transformation is applied to the predicted values and the corresponding interval bounds. The standard errors are omitted from the printed output. Also, vcov=TRUE is ignored when using the transf option.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. San Diego, CA: Academic Press.

Riley, R. D., Higgins, J. P. T., & Deeks, J. J. (2011). Interpretation of random effects meta-analyses. *British Medical Journal*, **342**, d549. https://doi.org/10.1136/bmj.d549

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

fitted.rma, blup.rma.uni

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit random-effects model
res <- rma(yi, vi, data=dat)

### average risk ratio with 95% CI
predict(res, transf=exp)

### fit mixed-effects model with absolute latitude as a moderator
res <- rma(yi, vi, mods = ~ ablat, data=dat)

### predicted average risk ratios for given absolute latitude values
predict(res, transf=exp, addx=TRUE)

### predicted average risk ratios for 10-60 degrees absolute latitude
predict(res, newmods=c(10, 20, 30, 40, 50, 60), transf=exp, addx=TRUE)

### fit mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)

### predicted average risk ratios for 10 and 60 degrees latitude in 1950 and 1980
predict(res, newmods=cbind(c(10,60,10,60),c(1950,1950,1980,1980)), transf=exp, addx=TRUE)

### fit mixed-effects model with two moderators (one of which is a factor)
res <- rma(yi, vi, mods = ~ ablat + factor(alloc), data=dat)

### examine how the factor was actually coded for the studies in the dataset
predict(res, addx=TRUE)

### predictd average risk ratios at 30 degrees for the three factor levels
### note: the contrast (dummy) variables need to specified explicitly here
predict(res, newmods=c(30, 0, 0), addx=TRUE)   # for alternate  allocation
predict(res, newmods=c(30, 1, 0), addx=TRUE)   # for random     allocation
predict(res, newmods=c(30, 0, 1), addx=TRUE)   # for systematic allocation

### can also use named vector with arbitrary order and abbreviated variable names
predict(res, newmods=c(sys=0, ran=0, abl=30))
predict(res, newmods=c(sys=0, ran=1, abl=30))
predict(res, newmods=c(sys=1, ran=0, abl=30))
```

print.anova.rma *Print Method for 'anova.rma' Objects*

Description

Print method for objects of class "anova.rma".

Usage

```
## S3 method for class 'anova.rma'
print(x, digits=x$digits, ...)
```

Arguments

`x` an object of class "anova.rma".

`digits` integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).

`...` other arguments.

Details

The output includes:

- the number of parameters in the full and the reduced model.
- the AIC, BIC, AICc, and log-likelihood of the full and the reduced model.
- the value of the likelihood ratio test statistic.
- the corresponding p-value.
- the test statistic of the test for (residual) heterogeneity for the full and the reduced model.
- the estimate of τ^2 from the full and the reduced model. Suppressed for fixed-effects models.
- R2amount (in percent) of heterogeneity in the reduced model that is accounted for in the full model (NA for fixed-effects models or for "rma.mv" objects). This can be regarded as a pseudo R^2 statistic (Raudenbush, 2009). Note that the value may not be very accurate unless k is large (Lopez-Lopez et al., 2014).

The last two items are not provided when comparing "rma.mv" models.

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

López-López, J. A., Marín-Martínez, F., Sánchez-Meca, J., Van den Noortgate, W., & Viechtbauer, W. (2014). Estimation of the predictive power of the model in mixed-effects meta-regression: A simulation study. *British Journal of Mathematical and Statistical Psychology*, **67**(1), 30–48. https://doi.org/10.1111/bmsp.12002

Raudenbush, S. W. (2009). Analyzing effect sizes: Random effects models. In H. Cooper, L. V. Hedges, & J. C. Valentine (Eds.), *The handbook of research synthesis and meta-analysis* (2nd ed., pp. 295–315). New York: Russell Sage Foundation.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`anova.rma`

`print.confint.rma` *Print Methods for 'confint.rma' and 'list.confint.rma' Objects*

Description

Print methods for objects of class `"confint.rma"` and `"list.confint.rma"`.

Usage

```
## S3 method for class 'confint.rma'
print(x, digits=x$digits, ...)
## S3 method for class 'list.confint.rma'
print(x, digits=x$digits, ...)
```

Arguments

`x`	an object of class `"confint.rma"` or `"list.confint.rma"`.
`digits`	integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
`...`	other arguments.

Details

The output includes:

- estimate of the model coefficient or variance/correlation parameter
- lower bound of the confidence interval
- upper bound of the confidence interval

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`confint.rma.uni`, `confint.rma.mv`

`print.escalc` *Print and Summary Methods for 'escalc' Objects*

Description

Print and summary methods for objects of class `"escalc"`.

Usage

```
## S3 method for class 'escalc'
print(x, digits=attr(x,"digits"), ...)

## S3 method for class 'escalc'
summary(object, out.names=c("sei","zi","pval","ci.lb","ci.ub"), var.names,
        H0=0, append=TRUE, replace=TRUE, level=95, olim, digits, transf, ...)
```

Arguments

`x`	an object of class `"escalc"`.
`object`	an object of class `"escalc"`.
`digits`	integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
`out.names`	character string with four elements to specify the variable names for the standard errors, test statistics, and lower/upper confidence interval bounds.
`var.names`	character string with two elements to specify the variable names for the observed effect sizes or outcomes and the sampling variances (the default is to take the value from the object if possible).
`H0`	numeric value to specify the value of the effect size or outcome under the null hypothesis (the default is 0).
`append`	logical to specify whether the data frame specified via the `object` argument should be returned together with the additional variables that are calculated by the `summary` function (the default is `TRUE`).
`replace`	logical to specify whether existing values for `sei`, `zi`, `ci.lb`, and `ci.ub` in the data frame should be replaced or not. Only relevant when the data frame already contains these variables. If `replace=TRUE` (the default), all of the existing values will be overwritten. If `replace=FALSE`, only `NA` values will be replaced.

`level`	numeric value between 0 and 100 to specify the confidence interval level (the default is 95).
`olim`	optional argument to specify observation/outcome limits. If unspecified, no limits are used.
`transf`	optional argument to specify a function that should be used to transform the observed effect sizes or outcomes and interval bounds (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used. Any additional arguments needed for the function specified here can be passed via `...`.
`...`	other arguments.

Value

The `print.escalc` function formats and prints the data frame, so that the observed effect sizes or outcomes and sampling variances are rounded (to the number of digits specified).

The `summary.escalc` function creates an object that is a data frame containing the original data (if `append=TRUE`) and the following components:

`yi`	observed effect sizes or outcomes (transformed if `transf` is specified).
`vi`	corresponding sampling variances.
`sei`	correponding standard errors.
`zi`	test statistics for testing H_0: $\theta_i = \mathrm{H0}$ (i.e., `(yi-H0)/sei`).
`pval`	corresponding p-values.
`ci.lb`	lower confidence interval bounds (transformed if `transf` is specified).
`ci.ub`	upper confidence interval bounds (transformed if `transf` is specified).

When the `transf` argument is specified, elements `vi`, `sei`, `zi`, and `pval` are not included (since these only apply to the untransformed effect sizes or outcomes).

Note that the actual variable names above depend on the `out.names` (and `var.names`) arguments. If the data frame already contains variables with names as specified by the `out.names` argument, the values for these variables will be overwritten when `replace=TRUE` (which is the default). By setting `replace=FALSE`, only values that are `NA` will be replaced.

The `print.escalc` function again formats and prints the data frame, rounding the added variables to the number of digits specified.

Note

If some transformation function has been specified for the `transf` argument, then `yi`, `ci.lb`, and `ci.ub` will be transformed accordingly. However, `vi` and `sei` then still reflect the sampling variances and standard errors of the untransformed values.

The `summary.escalc` function computes `level` % Wald-type confidence intervals, which may or may not be the most accurate method for computing confidence intervals for the chosen effect size or outcome measure.

If the outcome measure used is bounded (e.g., correlations are bounded between -1 and +1, proportions are bounded between 0 and 1), one can use the `olim` argument to enforce those observation/outcome limits (the observed outcomes and confidence intervals cannot exceed those bounds then).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

escalc

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)
dat

### apply summary function
summary(dat)
summary(dat, transf=exp)
```

---

print.fsn *Print Method for 'fsn' Objects*

Description

Print method for objects of class "fsn".

Usage

```
## S3 method for class 'fsn'
print(x, digits=x$digits, ...)
```

Arguments

x	an object of class "fsn".
digits	integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
...	other arguments.

Details

The output shows the results from the fail-safe N calculation.

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

fsn

---

print.gosh.rma *Print Method for 'gosh.rma' Objects*

Description

Print method for objects of class "gosh.rma".

Usage

```
## S3 method for class 'gosh.rma'
print(x, digits=x$digits, ...)
```

Arguments

x	an object of class "gosh.rma".
digits	integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
...	other arguments.

Details

The output shows how many model fits were attempted, how many succeeded, and summary statistics (i.e., the mean, minimum, first quartile, median, third quartile, and maximum) for the various measures of (residual) heterogeneity and the model coefficient(s) computed across all of the subsets.

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`gosh.rma`

`print.hc.rma.uni` *Print Method for 'hc.rma.uni' Objects*

Description

Print method for objects of class `"hc.rma.uni"`.

Usage

```
## S3 method for class 'hc.rma.uni'
print(x, digits=x$digits, ...)
```

Arguments

`x`	an object of class `"hc.rma.uni"`.
`digits`	integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
`...`	other arguments.

Details

The output is a data frame with two rows, the first (labeled `rma`) corresponding to the results based on the usual estimation method, the second (labeled `hc`) corresponding to the results based on the method by Henmi and Copas (2010). The data frame includes the following variables:

- the method used to estimate τ^2 (always `DL` for `hc`)
- the estimated amount of heterogeneity
- the estimated average true outcome
- the corresponding standard error (`NA` when `transf` argument has been used)
- the lower and upper confidence interval bounds

Value

The function returns the data frame invisibly.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`hc.rma.uni`

`print.list.rma` *Print method for 'list.rma' Objects*

Description

Print method for objects of class `"list.rma"`.

Usage

```
## S3 method for class 'list.rma'
print(x, digits=x$digits, ...)
```

Arguments

`x`	an object of class `"list.rma"`.
`digits`	integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
`...`	other arguments.

Value

See the documentation of the function that creates the `"list.rma"` object for details on what is printed. Regardless of what is printed, a data frame with the results is also returned invisibly.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

print.matreg *Print Method for 'matreg' Objects*

Description

Print method for objects of class "matreg".

Usage

```
## S3 method for class 'matreg'
print(x, digits=x$digits,
    signif.stars=getOption("show.signif.stars"), signif.legend=signif.stars, ...)
```

Arguments

x	an object of class "matreg".
digits	integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.
signif.stars	logical to specify whether p-values should be encoded visually with 'significance stars'. Defaults to the show.signif.stars slot of options.
signif.legend	logical to specify whether the legend for the 'significance stars' should be printed. Defaults to the value for signif.stars.
...	other arguments.

Details

The output is a table with the estimated coefficients, corresponding standard errors, test statistics, p-values, and confidence interval bounds.

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

See Also

matreg

print.permutest.rma.uni

Print Method for 'permutest.rma.uni' Objects

Description

Print method for objects of class "permutest.rma.uni".

Usage

```
## S3 method for class 'permutest.rma.uni'
print(x, digits=x$digits, signif.stars=getOption("show.signif.stars"),
      signif.legend=signif.stars, ...)
```

Arguments

x	an object of class "permutest.rma.uni".
digits	integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
signif.stars	logical to specify whether p-values should be encoded visually with 'significance stars'. Defaults to the show.signif.stars slot of options.
signif.legend	logical to specify whether the legend for the 'significance stars' should be printed. Defaults to the value for signif.stars.
...	other arguments.

Details

The output includes:

- the results of the omnibus test of moderators. Suppressed if the model includes only one coefficient (e.g., only an intercept, like in the fixed- and random-effects model). The p-value based on the permutation test is indicated by p-val*.
- a table with the estimated coefficients, corresponding standard errors, test statistics, p-values, and confidence interval bounds. The p-values based on the permutation tests are indicated by pval*. When permutation-based CIs have been obtained, then the CI bounds are indicated with ci.lb* and ci.ub*.

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`permutest.rma.uni`

`print.ranktest` *Print Method for 'ranktest' Objects*

Description

Print method for objects of class `"ranktest"`.

Usage

```
## S3 method for class 'ranktest'
print(x, digits=x$digits, ...)
```

Arguments

`x`	an object of class `"ranktest"`.
`digits`	integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
`...`	other arguments.

Details

The output includes:

- the estimated value of Kendall's tau rank correlation coefficient
- the corresponding p-value for the test that the true tau is equal to zero

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

ranktest

---

print.regtest *Print Method for 'regtest' Objects*

Description

Print method for objects of class "regtest".

Usage

```
## S3 method for class 'regtest'
print(x, digits=x$digits, ret.fit=x$ret.fit, ...)
```

Arguments

x	an object of class "regtest".
digits	integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
ret.fit	logical to specify whether the full results from the fitted model should also be returned. If unspecified, the default is to take the value from the object.
...	other arguments.

Details

The output includes:

- the model used for the regression test
- the predictor used for the regression test
- the results from the fitted model (only when ret.fit=TRUE)
- the test statistic of the test that the predictor is unreleated to the outcomes
- the degrees of freedom of the test statistic (only if the test statistic follows a t-distribution)
- the corresponding p-value
- the 'limit estimate' and its corresponding CI (only for predictors "sei" "vi", "ninv", or "sqrtninv" and when the model does not contain any additional moderators)

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

regtest

print.rma *Print and Summary Methods for 'rma' Objects*

Description

Print and summary methods for objects of class "rma.uni", "rma.mh", "rma.peto", "rma.glmm", "rma.glmm", and "rma.mv".

Usage

```
## S3 method for class 'rma.uni'
print(x, digits, showfit=FALSE, signif.stars=getOption("show.signif.stars"),
      signif.legend=signif.stars, ...)

## S3 method for class 'rma.mh'
print(x, digits, showfit=FALSE, ...)

## S3 method for class 'rma.peto'
print(x, digits, showfit=FALSE, ...)

## S3 method for class 'rma.glmm'
print(x, digits, showfit=FALSE, signif.stars=getOption("show.signif.stars"),
      signif.legend=signif.stars, ...)

## S3 method for class 'rma.mv'
print(x, digits, showfit=FALSE, signif.stars=getOption("show.signif.stars"),
      signif.legend=signif.stars, ...)

## S3 method for class 'rma'
summary(object, digits, showfit=TRUE, ...)

## S3 method for class 'summary.rma'
print(x, digits, showfit=TRUE, signif.stars=getOption("show.signif.stars"),
      signif.legend=signif.stars, ...)
```

Arguments

`x` an object of class `"rma.uni"`, `"rma.mh"`, `"rma.peto"`, `"rma.glmm"`, `"rma.mv"`, or `"summary.rma"` (for `print`).

`object` an object of class `"rma"` (for `summary`).

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.

`showfit` logical to specify whether the fit statistics and information criteria should be printed (the default is `FALSE` for `print` and `TRUE` for `summary`).

`signif.stars` logical to specify whether p-values should be encoded visually with 'significance stars'. Defaults to the `show.signif.stars` slot of `options`.

`signif.legend` logical to specify whether the legend for the 'significance stars' should be printed. Defaults to the value for `signif.stars`.

`...` other arguments.

Details

The output includes:

- the log-likelihood, deviance, AIC, BIC, and AICc value (when setting `showfit=TRUE` or by default for `summary`).
- for objects of class `"rma.uni"` and `"rma.glmm"`, the amount of (residual) heterogeneity in the random/mixed-effects model (i.e., the estimate of τ^2 and its square root). Suppressed for fixed-effects models. The (asymptotic) standard error of the estimate of τ^2 is also provided (where possible).
- for objects of `"rma.mv"`, a table providing information about the variance components and correlations in the model. For σ^2 components, the estimate and its square root are provided, in addition to the number of values/levels, whether the component was fixed or estimated, and the name of the grouping variable/factor. If the `R` argument was used to specify known correlation matrices, this is also indicated. For models with an '`~ inner | outer`' formula term, the name of the inner and outer grouping variable/factor are given and the number of values/levels of these variables/factors. In addition, for each τ^2 component, the estimate and its square root are provided, the number of effects or outcomes observed at each level of the inner grouping variable/factor (only for `struct="HCS"`, `struct="DIAG"`, `struct="HAR"`, and `struct="UN"`), and whether the component was fixed or estimated. Finally, either the estimate of ρ (for `struct="CS"`, `struct="AR"`, `struct="CAR"`, `struct="HAR"`, or `struct="HCS"`) or the entire estimated correlation matrix (for `struct="UN"`) between the levels of the inner grouping variable/factor is provided, again with information whether a particular correlation was fixed or estimated, and how often each combination of levels of the inner grouping variable/factor was observed across the levels of the outer grouping variable/factor. If there is a second '`~ inner | outer`' formula term, the same information as described above will be provided, but now for the γ^2 and ϕ components.
- the I^2 statistic, which estimates (in percent) how much of the total variability in the observed effect sizes or outcomes (which is composed of heterogeneity plus sampling variability) can be attributed to heterogeneity among the true effects. For a meta-regression model, I^2 estimates how much of the unaccounted variability (which is composed of residual heterogeneity plus

sampling variability) can be attributed to residual heterogeneity. See 'Note' for how I^2 is computed.

- the H^2 statistic, which estimates the ratio of the total amount of variability in the observed effect sizes or outcomes to the amount of sampling variability. For a meta-regression model, H^2 estimates the ratio of the unaccounted variability in the observed effect sizes or outcomes to the amount of sampling variability. See 'Note' for how H^2 is computed.
- for objects of class `"rma.uni"`, the R^2 statistic, which estimates the amount of heterogeneity accounted for by the moderators included in the model and can be regarded as a pseudo R^2 statistic (Raudenbush, 2009). Only provided when fitting a mixed-effects models (i.e., for models including moderators). This is suppressed (and set to `NULL`) for models without moderators, fixed-effects models, or if the model does not contain an intercept. See 'Note' for how R^2 is computed.
- for objects of class `"rma.glmm"`, the amount of study level variability (only when using a model that models study level differences as a random effect).
- the results of the test for (residual) heterogeneity. This is the usual Q-test for heterogeneity when not including moderators in the model and the Q_E-test for residual heterogeneity when moderators are included. For objects of class `"rma.glmm"`, the results from a Wald-type test and a likelihood ratio test are provided (see `rma.glmm` for more details).
- the results of the omnibus (Wald-type) test of the coefficients in the model (the indices of the coefficients tested are also indicated). Suppressed if the model includes only one coefficient (e.g., only an intercept, like in the fixed- and random-effects model).
- a table with the estimated coefficients, corresponding standard errors, test statistics, p-values, and confidence interval bounds.
- the Cochran-Mantel-Haenszel test and Tarone's test for heterogeneity (only when analyzing odds ratios using the Mantel-Haenszel method, i.e., `"rma.mh"`).

Value

The `print` functions do not return an object. The `summary` function returns the object passed to it (with additional class `"summary.rma"`).

Note

For random-effects models, the I^2 statistic is computed with

$$I^2 = 100\% \times \frac{\hat{\tau}^2}{\hat{\tau}^2 + \tilde{v}},$$

where $\hat{\tau}^2$ is the estimated value of τ^2 and

$$\tilde{v} = \frac{(k-1)\sum w_i}{(\sum w_i)^2 - \sum w_i^2},$$

where $w_i = 1/v_i$ is the inverse of the sampling variance of the ith study ($\tilde{v}$ is equation 9 in Higgins & Thompson, 2002, and can be regarded as the 'typical' within-study variance of the observed effect sizes or outcomes). The H^2 statistic is computed with

$$H^2 = \frac{\hat{\tau}^2 + \tilde{v}}{\tilde{v}}.$$

Analogous equations are used for mixed-effects models.

Therefore, depending on the estimator of τ^2 used, the values of I^2 and H^2 will change. For random-effects models, I^2 and H^2 are often computed with $I^2 = (Q - (k-1))/Q$ and $H^2 = Q/(k-1)$, where Q denotes the statistic of the test for heterogeneity and k the number of studies (i.e., observed effect sizes or outcomes) included in the meta-analysis. The equations used in the metafor package to compute these statistics are more general and have the advantage that the values of I^2 and H^2 will be consistent with the estimated value of τ^2 (i.e., if $\hat{\tau}^2 = 0$, then $I^2 = 0$ and $H^2 = 1$ and if $\hat{\tau}^2 > 0$, then $I^2 > 0$ and $H^2 > 1$).

The two definitions of I^2 and H^2 actually coincide when using the DerSimonian-Laird estimator of τ^2 (i.e., the commonly used equations are actually special cases of the more general definitions given above). Therefore, if you prefer the more conventional definitions of these statistics, use `method="DL"` when fitting the random/mixed-effects model with the `rma.uni` function. The conventional definitions are also automatically used when fitting fixed-effects models.

The pseudo R^2 statistic (Raudenbush, 2009) is computed with

$$R^2 = \frac{\hat{\tau}^2_{RE} - \hat{\tau}^2_{ME}}{\hat{\tau}^2_{RE}},$$

where $\hat{\tau}^2_{RE}$ denotes the estimated value of τ^2 based on the random-effects model (i.e., the total amount of heterogeneity) and $\hat{\tau}^2_{ME}$ denotes the estimated value of τ^2 based on the mixed-effects model (i.e., the residual amount of heterogeneity). It can happen that $\hat{\tau}^2_{RE} < \hat{\tau}^2_{ME}$, in which case R^2 is set to zero (and also if $\hat{\tau}^2_{RE} = 0$). Again, the value of R^2 will change depending on the estimator of τ^2 used. This statistic is only computed when the mixed-effects model includes an intercept (so that the random-effects model is clearly nested within the mixed-effects model). You can also use the `anova.rma` function to compute R^2 for any two models that are known to be nested. Note that the pseudo R^2 statistic may not be very accurate unless k is large (Lopez-Lopez et al., 2014).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Higgins, J. P. T., & Thompson, S. G. (2002). Quantifying heterogeneity in a meta-analysis. *Statistics in Medicine*, **21**(11), 1539–1558. https://doi.org/10.1002/sim.1186

López-López, J. A., Marín-Martínez, F., Sánchez-Meca, J., Van den Noortgate, W., & Viechtbauer, W. (2014). Estimation of the predictive power of the model in mixed-effects meta-regression: A simulation study. *British Journal of Mathematical and Statistical Psychology*, **67**(1), 30–48. https://doi.org/10.1111/bmsp.12002

Raudenbush, S. W. (2009). Analyzing effect sizes: Random effects models. In H. Cooper, L. V. Hedges, & J. C. Valentine (Eds.), *The handbook of research synthesis and meta-analysis* (2nd ed., pp. 295–315). New York: Russell Sage Foundation.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`rma.uni`, `rma.mh`, `rma.peto`, `rma.glmm`, `rma.mv`

print.robust.rma *Print Method for 'robust.rma' Objects*

Description

Print method for objects of class "robust.rma".

Usage

```
## S3 method for class 'robust.rma'
print(x, digits=x$digits, signif.stars=getOption("show.signif.stars"),
      signif.legend=signif.stars, ...)
```

Arguments

x	an object of class "robust.rma".
digits	integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
signif.stars	logical to specify whether p-values should be encoded visually with 'significance stars'. Defaults to the show.signif.stars slot of options.
signif.legend	logical to specify whether the legend for the 'significance stars' should be printed. Defaults to the value for signif.stars.
...	other arguments.

Details

The output includes:

- information on the number of observed effect sizes or outcomes, number of clusters, and the number of effect sizes or outcomes per cluster.
- the results of the omnibus (Wald-type) test of the coefficients in the model (the indices of the coefficients tested are also indicated). Suppressed if the model includes only one coefficient (e.g., only an intercept, like in the fixed- and random-effects model).
- a table with the estimated coefficients, corresponding standard errors, test statistics, p-values, and confidence interval bounds.

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

robust.rma.uni, robust.rma.mv

---

profile.rma *Profile Plots for 'rma' Objects*

Description

Function to profile the (restricted) log-likelihood for objects of class "rma.uni", "rma.mv", "rma.uni.selmodel", and "rma.ls".

Usage

```
## S3 method for class 'rma.uni'
profile(fitted, xlim, ylim, steps=20,
        lltol=1e-03, progbar=TRUE, parallel="no", ncpus=1, cl=NULL,
        plot=TRUE, pch=19, cline=FALSE, ...)

## S3 method for class 'rma.mv'
profile(fitted, sigma2, tau2, rho, gamma2, phi,
        xlim, ylim, steps=20, lltol=1e-03, progbar=TRUE,
        parallel="no", ncpus=1, cl=NULL,
        plot=TRUE, pch=19, cline=FALSE, ...)

## S3 method for class 'rma.uni.selmodel'
profile(fitted, tau2, delta,
        xlim, ylim, steps=20, lltol=1e-03, progbar=TRUE,
        parallel="no", ncpus=1, cl=NULL,
        plot=TRUE, pch=19, cline=FALSE, ...)

## S3 method for class 'rma.ls'
profile(fitted, alpha, xlim, ylim, steps=20,
        lltol=1e-03, progbar=TRUE, parallel="no", ncpus=1, cl=NULL,
        plot=TRUE, pch=19, cline=FALSE, ...)

## S3 method for class 'profile.rma'
print(x, ...)
## S3 method for class 'profile.rma'
plot(x, xlim, ylim, pch=19,
     xlab, ylab, main, cline=FALSE, ...)
```

Arguments

`fitted` an object of class "rma.uni", "rma.mv", "rma.uni.selmodel", or "rma.ls".

`x` an object of class "profile.rma" (for plot and print).

`sigma2` optional integer to specify for which σ^2 value the likelihood should be profiled.

`tau2` optional integer to specify for which τ^2 value the likelihood should be profiled.

`rho` optional integer to specify for which ρ value the likelihood should be profiled.

`gamma2` optional integer to specify for which γ^2 value the likelihood should be profiled.

`phi` optional integer to specify for which ϕ value the likelihood should be profiled.

`delta` optional integer to specify for which δ value the likelihood should be profiled.

`alpha` optional integer to specify for which α value the likelihood should be profiled.

`xlim` optional vector to specify the lower and upper limit of the parameter over which the profiling should be done. If unspecified, the function tries to set these limits automatically.

`ylim` optional vector to specify the y-axis limits when plotting the profiled likelihood. If unspecified, the function tries to set these limits automatically.

`steps` number of points between `xlim[1]` and `xlim[2]` (inclusive) for which the likelihood should be evaluated (the default is 20).

`lltol` numerical tolerance used when comparing values of the profiled log-likelihood with the log-likelihood of the fitted model (the default is 1e-03).

`progbar` logical to specify whether a progress bar should be shown (the default is `TRUE`).

`parallel` character string to specify whether parallel processing should be used (the default is "no"). For parallel processing, set to either "snow" or "multicore". See 'Details'.

`ncpus` integer to specify the number of processes to use in the parallel processing.

`cl` optional cluster to use if `parallel="snow"`. If not supplied, a cluster on the local machine is created for the duration of the call.

`plot` logical to specify whether the profile plot should be drawn after profiling is finished (the default is `TRUE`).

`pch` plotting symbol to use. By default, a filled circle is used. See `points` for other options.

`cline` logical to specify whether a horizontal reference line should be added to the plot that indicates the log-likelihood value corresponding to the 95% profile confidence interval (the default is `FALSE`).

`xlab` title for the x-axis. If unspecified, the function tries to set an appropriate axis title.

`ylab` title for the y-axis. If unspecified, the function tries to set an appropriate axis title.

`main` title for the plot. If unspecified, the function tries to set an appropriate title.

`...` other arguments.

Details

The function fixes a particular parameter of the model and then computes the maximized (restricted) log-likelihood over the remaining parameters of the model. By doing this for a range of values for the parameter that was fixed, a profile of the (restricted) log-likelihood is constructed.

The parameter(s) that can be profiled depend on the model object:

- For objects of class "rma.uni" obtained with the rma.uni function, the function profiles over parameter τ^2 (not for fixed-effects models).
- For objects of class "rma.mv" obtained with the rma.mv function, profiling is done by default over all (non-fixed) variance and correlation components of the model. Alternatively, one can use the sigma2, tau2, rho, gamma2, or phi arguments to specify over which parameter the profiling should be done. Only one of these arguments can be used at a time. A single integer is used to specify the number of the parameter.
- For selection model objects of class "rma.uni.selmodel" obtained with the selmodel.rma.uni function, profiling is done by default over τ^2 (for models where this is an estimated parameter) and all (non-fixed) selection model parameters. Alternatively, one can choose to profile only τ^2 by setting tau2=TRUE or one can select one of the selection model parameters to profile by specifying its number via the delta argument.
- For location-scale model objects of class "rma.ls" obtained with the rma.uni function, profiling is done by default over all (non-fixed) α parameters that are part of the scale model.

A profile plot should show a single peak at the corresponding ML/REML estimate (assuming that the model was fitted with ML/REML estimation). The value of the parameter estimate is indicated by a dotted vertical line and its log-likelihood value by a dotted horizontal line. Hence, the intersection of these two lines should correspond to the peak.

When profiling a variance component (or some other parameter that cannot be negative), the peak may be at zero (if this corresponds to the ML/REML estimate of the parameter). In this case, the profiled log-likelihood should be a monotonically decreasing function of the parameter.

If the profiled log-likelihood has multiple peaks, this indicates that the likelihood surface is not unimodal. In such cases, the ML/REML estimate may correspond to a local optimum (when the intersection of the two dotted lines is not at the highest peak).

If the profile is flat (over the entire parameter space or large portions of it), then this suggests that at least some of the parameters of the model are not identifiable (and the parameter estimates obtained are to some extent arbitrary). See Raue et al. (2009) for some further discussion of parameter identifiability (structurally and practically) and the use of profile likelihoods to check for this.

The function checks whether any profiled log-likelihood value is actually larger than the log-likelihood of the fitted model (using a numerical tolerance of lltol). If so, a warning is issued as this might indicate that the optimizer did not identify the actual ML/REML estimate.

Profiling requires repeatedly refitting the model, which can be slow when k is large and/or the model is complex (the latter especially applies to "rma.mv" objects and also to certain "rma.uni.selmodel" or "rma.ls" objects). On machines with multiple cores, one can usually speed things up by delegating the model fitting to separate worker processes, that is, by setting parallel="snow" or parallel="multicore" and ncpus to some value larger than 1. Parallel processing makes use of the parallel package, using the makePSOCKcluster and parLapply functions when parallel="snow" or using mclapply when parallel="multicore" (the latter only works on Unix/Linux-alikes). With parallel::detectCores(), one can check on the number of available cores on the local machine.

Value

An object of class "profile.rma". The object is a list (or list of lists) containing the following components:

`sigma2` values of σ^2 over which the likelihood was profiled (only when profiling was actually done over σ^2).

`tau2` values of τ^2 over which the likelihood was profiled (only when profiling was actually done over τ^2).

`rho` values of ρ over which the likelihood was profiled (only when profiling was actually done over ρ).

`gamma2` values of γ^2 over which the likelihood was profiled (only when profiling was actually done over γ^2).

`phi` values of ϕ over which the likelihood was profiled (only when profiling was actually done over ϕ).

`delta` values of δ over which the likelihood was profiled (only when profiling was actually done over δ).

`alpha` values of α over which the likelihood was profiled (only when profiling was actually done over α).

`ll` (restricted) log-likelihood at the corresponding parameter value.

`beta` a matrix with the estimated model coefficients at the corresponding parameter value.

`ci.lb` a matrix with the lower confidence interval bounds for the model coefficients at the corresponding parameter value.

`ci.ub` a matrix with the upper confidence interval bounds for the model coefficients at the corresponding parameter value.

`...` some additional elements/values.

Note that the list is returned invisibly.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Klingmuller, U., & Timmer, J. (2009). Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, **25**(15), 1923–1929. https://doi.org/10.1093/bioinformatics/btp358

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mv, selmodel.rma.uni, confint.rma.uni, confint.rma.mv, confint.rma.uni.selmodel

Examples

```
### calculate log odds ratios and corresponding sampling variances
dat <- escalc(measure=”OR”, ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit random-effects model using rma.uni()
res <- rma(yi, vi, data=dat)

### profile over tau^2
profile(res, progbar=FALSE)

### change data into long format
dat.long <- to.long(measure=”OR”, ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### set levels of group variable (”exp” = experimental/vaccinated; ”con” = control/non-vaccinated)
levels(dat.long$group) <- c(”exp”, ”con”)

### set ”con” to reference level
dat.long$group <- relevel(dat.long$group, ref=”con”)

### calculate log odds and corresponding sampling variances
dat.long <- escalc(measure=”PLO”, xi=out1, mi=out2, data=dat.long)

### fit bivariate random-effects model using rma.mv()
res <- rma.mv(yi, vi, mods = ~ group, random = ~ group | study, struct=”UN”, data=dat.long)
res

### profile over tau^2_1, tau^2_2, and rho
### note: for rho, adjust region over which profiling is done ('zoom in' on area around estimate)
## Not run:
par(mfrow=c(3,1))
profile(res, tau2=1)
profile(res, tau2=2)
profile(res, rho=1, xlim=c(.90, .98))
## End(Not run)

### an example where the peak is at 0
dat <- escalc(measure=”RD”, n1i=n1i, n2i=n2i, ai=ai, ci=ci, data=dat.hine1989)
res <- rma(yi, vi, data=dat)
par(mfrow=c(1,1))
profile(res, progbar=FALSE)
```

`qqnorm.rma` *Normal QQ Plots for ’rma’ Objects*

Description

Function to create normal QQ plots for objects of class "rma.uni", "rma.mh", and "rma.peto".

Usage

```
## S3 method for class 'rma.uni'
qqnorm(y, type="rstandard", pch=19, envelope=TRUE,
       level=y$level, bonferroni=FALSE, reps=1000, smooth=TRUE, bass=0,
       label=FALSE, offset=0.3, pos=13, lty, ...)
## S3 method for class 'rma.mh'
qqnorm(y, type="rstandard", pch=19, label=FALSE, offset=0.3, pos=13, ...)
## S3 method for class 'rma.peto'
qqnorm(y, type="rstandard", pch=19, label=FALSE, offset=0.3, pos=13, ...)
## S3 method for class 'rma.glmm'
qqnorm(y, ...)
## S3 method for class 'rma.mv'
qqnorm(y, ...)
```

Arguments

`y` an object of class `"rma.uni"`, `"rma.mh"`, or `"rma.peto"`. The method is not yet implemented for objects of class `"rma.glmm"` or `"rma.mv"`.

`type` character string (either `"rstandard"` (default) or `"rstudent"`) to specify whether standardized residuals or studentized deleted residuals should be used in creating the plot. See ‘Details’.

`pch` plotting symbol to use for the observed outcomes. By default, a filled circle is used. See `points` for other options.

`envelope` logical to specify whether a pseudo confidence envelope should be simulated and added to the plot (the default is `TRUE`)). Only for objects of class `"rma.uni"`. See ‘Details’.

`level` numeric value between 0 and 100 to specify the level of the pseudo confidence envelope (the default is to take the value from the object).

`bonferroni` logical to specify whether the bounds of the envelope should be Bonferroni corrected.

`reps` numeric value to specify the number of iterations to use for simulating the pseudo confidence envelope (the default is 1000).

`smooth` logical to specify whether the results from the simulation should be smoothed (the default is `TRUE`).

`bass` numeric value that controls the degree of smoothing (the default is 0).

`label` argument to control the labeling of the points (the default is `FALSE`). See ‘Details’.

`offset` argument to control the distance between the points and the corresponding labels.

`pos` argument to control the position of the labels.

`lty` optional character string to specify the line type for the diagonal line and the pseudo confidence envelope. If unspecified, the function sets this to `c("solid","dotted")` by default.

`...` other arguments.

Details

The plot shows the theoretical quantiles of a normal distribution on the horizontal axis against the observed quantiles for either the standardized residuals (`type="rstandard"`, the default) or the externally standardized residuals (`type="rstudent"`) on the vertical axis (see `residuals.rma` for details on the definition of these residual types).

For reference, a line is added to the plot with slope of 1, going through the (0,0) point.

For objects of class `"rma.uni"`, it is also possible to add a pseudo confidence envelope to the plot. The envelope is created based on the quantiles of sets of pseudo residuals simulated from the given model (for details, see Cook & Weisberg, 1982). The number of sets simulated can be controlled with the `reps` argument. When `smooth=TRUE`, the simulated bounds are smoothed with Friedman's SuperSmoother (see `supsmu`). The `bass` argument can be set to a number between 0 and 10, with higher numbers indicating increasing smoothness. If `bonferroni=TRUE`, the envelope bounds are Bonferroni corrected, so that the envelope can be regarded as a confidence region for all k residuals simultaneously. The default however is `bonferroni=FALSE`, which makes the plot more sensitive to deviations from normality.

With the `label` argument, one can control whether points in the plot will be labeled (e.g., to identify outliers). If `label="all"` (or `label=TRUE`), all points in the plot will be labeled. If `label="out"`, points falling outside of the confidence envelope will be labeled (only available for objects of class `"rma.uni"`). Finally, one can also set this argument to a numeric value (between 1 and k), indicating how many of the most extreme points should be labeled (for example, with `label=1` only the most extreme point would be labeled, while with `label=3`, the most extreme, and the second and third most extreme points would be labeled). With the `offset` argument, one can adjust the distance between the labels and the corresponding points. The `pos` argument is the position specifier for the labels (`1`, `2`, `3`, and `4`, respectively indicate positions below, to the left of, above, and to the right of the points; `13` places the labels below the points for points that fall below the reference line and above otherwise; `24` places the labels to the left of the points for points that fall above the reference line and to the right otherwise).

Value

A list with components:

`x`	the x-axis coordinates of the points that were plotted.
`y`	the y-axis coordinates of the points that were plotted.

Note that the list is returned invisibly.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Cook, R. D., & Weisberg, S. (1982). *Residuals and influence in regression*. London: Chapman and Hall.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Wang, M. C., & Bushman, B. J. (1998). Using the normal quantile plot to explore meta-analytic data sets. *Psychological Methods*, **3**(1), 46–54. https://doi.org/10.1037/1082-989X.3.1.46

See Also

rma.uni, rma.mh, rma.peto, rma.glmm, rma.mv

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit random-effects model
res <- rma(yi, vi, data=dat)

### draw QQ plot
qqnorm(res)

### fit mixed-effects model with absolute latitude as moderator
res <- rma(yi, vi, mods = ~ ablat, data=dat)

### draw QQ plot
qqnorm(res)
```

radial *Radial (Galbraith) Plots for 'rma' Objects*

Description

Function to create radial (also called Galbraith) plots for objects of class "rma".

Usage

```
radial(x, ...)
galbraith(x, ...)

## S3 method for class 'rma'
radial(x, center=FALSE, xlim, zlim, xlab, zlab,
       atz, aty, steps=7, level=x$level, digits=2, back="lightgray",
       transf, targs, pch=19, arc.res=100, cex, ...)
```

Arguments

`x` an object of class "rma".

`center` logical to indicate whether the plot should be centered horizontally at the model estimate (the default is FALSE).

`xlim` x-axis limits. If unspecified, the function tries to set the x-axis limits to some sensible values.

`zlim` z-axis limits. If unspecified, the function tries to set the z-axis limits to some sensible values (note that the z-axis limits are the actual vertical limit of the plotting region).

`xlab` title for the x-axis. If unspecified, the function tries to set an appropriate axis title.

`zlab` title for the z-axis. If unspecified, the function tries to set an appropriate axis title.

`atz` position for the z-axis tick marks and labels. If unspecified, these values are set by the function.

`aty` position for the y-axis tick marks and labels. If unspecified, these values are set by the function.

`steps` the number of tick marks for the y-axis (the default is 7). Ignored when argument `aty` is used.

`level` numeric value between 0 and 100 to specify the level of the z-axis error region (the default is to take the value from the object).

`digits` integer to specify the number of decimal places to which the tick mark labels of the y-axis should be rounded (the default is 2).

`back` color of the z-axis error region. Set to `NA` to suppress shading of the region.

`transf` optional argument to specify a function that should be used to transform the y-axis labels (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified via `transf`.

`pch` plotting symbol. By default, a filled circle is used. See `points` for other options.

`arc.res` integer to specify the number of line segments to use when drawing the y-axis and confidence interval arcs (the default is 100).

`cex` optional character and symbol expansion factor. If unspecified, the function tries to set this to a sensible value.

`...` other arguments.

Details

For a fixed-effects model, the plot shows the inverse of the standard errors on the horizontal axis against the observed effect sizes or outcomes standardized by their corresponding standard errors on the vertical axis. Since the vertical axis corresponds to standardized values, it is referred to as the z-axis within this function. On the right hand side of the plot, an arc is drawn (referred to as the y-axis within this function) corresponding to the observed effect sizes or outcomes. A line projected from (0,0) through a particular point within the plot onto this arc indicates the value of the observed effect size or outcome for that point.

For a random-effects model, the function uses $1/\sqrt{v_i + \tau^2}$ for the horizontal axis, where v_i is the sampling variance of the observed effect size or outcome and τ^2 is the amount of heterogeneity as estimated based on the model. For the z-axis, $\sqrt{v_i + \tau^2}$ is used to standardize the observed effect sizes or outcomes.

If the model contains moderators, the function returns an error.

Value

A data frame with components:

x	the x-axis coordinates of the points that were plotted.
y	the y-axis coordinates of the points that were plotted.
ids	the study id numbers.
slab	the study labels.

Note that the data frame is returned invisibly.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Galbraith, R. F. (1988). Graphical display of estimates having differing standard errors. *Technometrics*, **30**(3), 271–281. https://doi.org/10.1080/00401706.1988.10488400

Galbraith, R. F. (1988). A note on graphical presentation of estimated odds ratios from several clinical trials. *Statistics in Medicine*, **7**(8), 889–894. https://doi.org/10.1002/sim.4780070807

Galbraith, R. F (1994). Some applications of radial plots. *Journal of the American Statistical Association*, **89**(428), 1232–1242. https://doi.org/10.1080/01621459.1994.10476864

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mh, rma.peto, rma.glmm, rma.mv

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)
dat

### fixed-effects model
res <- rma(yi, vi, data=dat, method="FE")

### draw radial plot
radial(res)

### line from (0,0) with slope equal to the log risk ratio from the 4th study
abline(a=0, b=dat$yi[4], lty="dotted")

### meta-analysis of the log risk ratios using a random-effects model
res <- rma(yi, vi, data=dat)

### draw radial plot
radial(res)
```

ranef *Best Linear Unbiased Predictions for 'rma.uni' and 'rma.mv' Objects*

Description

The function calculates best linear unbiased predictions (BLUPs) of the random effects for objects of class "rma.uni" and "rma.mv". Corresponding standard errors and prediction interval bounds are also provided.

Usage

```
## S3 method for class 'rma.uni'
ranef(object, level, digits, transf, targs, ...)
## S3 method for class 'rma.mv'
ranef(object, level, digits, transf, targs, verbose=FALSE, ...)
```

Arguments

`object` an object of class "rma.uni" or "rma.mv".

`level` numeric value between 0 and 100 to specify the prediction interval level. If unspecified, the default is to take the value from the object.

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.

`transf` optional argument to specify a function that should be used to transform the predicted values and interval bounds (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified under `transf`.

`verbose` logical to specify whether output should be generated on the progress of the computations (the default is `FALSE`).

`...` other arguments.

Value

For objects of class "rma.uni", an object of class "list.rma". The object is a list containing the following components:

`pred` predicted values.

`se` corresponding standard errors.

`pi.lb` lower bound of the prediction intervals.

`pi.ub` upper bound of the prediction intervals.

`...` some additional elements/values.

The object is formatted and printed with `print.list.rma`.

For objects of class "rma.mv", a list of data frames with the same components as described above.

Note

For best linear unbiased predictions that combine the fitted values based on the fixed effects and the estimated contributions of the random effects, see blup.

For predicted/fitted values that are based only on the fixed effects of the model, see fitted.rma and predict.rma.

Fixed-effects models (with or without moderators) do not contain random study effects. The BLUPs for these models will therefore be 0.

When using the transf argument, the transformation is applied to the predicted values and the corresponding interval bounds. The standard errors are then set equal to NA and are omitted from the printed output.

By default, a standard normal distribution is used to calculate the prediction intervals. When the model was fitted with test="t" or test="knha", then a t-distribution with $k-p$ degrees of freedom is used.

To be precise, it should be noted that the function actually calculates empirical BLUPs (eBLUPs), since the predicted values are a function of the estimated variance component(s).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Kackar, R. N., & Harville, D. A. (1981). Unbiasedness of two-stage estimation and prediction procedures for mixed linear models. Communications in Statistics, Theory and Methods, **10**(13), 1249–1261. https://doi.org/10.1080/03610928108828108

Raudenbush, S. W., & Bryk, A. S. (1985). Empirical Bayes meta-analysis. *Journal of Educational Statistics*, **10**(2), 75–98. https://doi.org/10.3102/10769986010002075

Robinson, G. K. (1991). That BLUP is a good thing: The estimation of random effects. *Statistical Science*, **6**(1), 15–32. https://doi.org/10.1214/ss/1177011926

Searle, S. R., Casella, G., & McCulloch, C. E. (1992). *Variance components*. Hoboken, NJ: Wiley.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mv, predict.rma, fitted.rma, blup.rma.uni

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### meta-analysis of the log risk ratios using a random-effects model
res <- rma(yi, vi, data=dat)

### BLUPs of the random effects
ranef(res)
```

`ranktest` *Rank Correlation Test for Funnel Plot Asymmetry*

Description

The function can be used to carry out the rank correlation test for funnel plot asymmetry.

Usage

```
ranktest(x, ...)

## S3 method for class 'rma'
ranktest(x, digits, ...)

## Default S3 method:
ranktest(x, vi, sei, subset, digits, ...)
```

Arguments

`x`	an object of class `"rma"` or a vector with the observed effect sizes or outcomes.
`vi`	vector with the corresponding sampling variances (needed if `x` is a vector with the observed effect sizes or outcomes).
`sei`	vector with the corresponding standard errors (note: only one of the two, `vi` or `sei`, needs to be specified).
`subset`	optional (logical or numeric) vector to specify the subset of studies that should be included in the test. Only relevant when passing a vector via `x`.
`digits`	integer to specify the number of decimal places to which the printed results should be rounded (the default is 4).
`...`	other arguments.

Details

The function carries out the rank correlation test as described by Begg and Mazumdar (1994). The test can be used to examine whether the observed effect sizes or outcomes and the corresponding sampling variances are correlated. A high correlation would indicate that the funnel plot is asymmetric, which may be a result of publication bias.

One can either pass an object of class `"rma"` to the function or a vector with the observed effect sizes or outcomes (via `x`) and the corresponding sampling variances via `vi` (or the standard errors via `sei`).

Value

An object of class `"ranktest"`. The object is a list containing the following components:

`tau`	the estimated value of Kendall's tau rank correlation coefficient
`pval`	the corresponding p-value for the test that the true tau is equal to zero

The results are formatted and printed with the `print.ranktest` function.

Note

The method does not depend on the model fitted. Therefore, regardless of the model passed to the function, the results of the rank test will always be the same. See regtest for tests of funnel plot asymmetry that are based on regression models and model dependent.

The function makes use of the cor.test function with method="kendall". If possible, an exact p-value is provided; otherwise, a large-sample approximation is used.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Begg, C. B., & Mazumdar, M. (1994). Operating characteristics of a rank correlation test for publication bias. *Biometrics*, **50**(4), 1088–1101. https://doi.org/10.2307/2533446

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

regtest

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit random-effects model
res <- rma(yi, vi, data=dat)

### carry out the rank correlation test
ranktest(res)

### can also pass the observed outcomes and corresponding sampling variances to the function
ranktest(dat$yi, dat$vi)
```

---

rcalc *Calculate the Variance-Covariance of Correlation Coefficients*

Description

The function can be used to calculate the variance-covariance matrix of correlation coefficients computed based on the same sample of subjects.

Usage

```
rcalc(x, ni, data, rtoz=FALSE, nfun="min", sparse=FALSE, ...)
```

Arguments

`x` a formula of the form `ri ~ var1 + var2 | study`. See 'Details'.

`ni` vector to specify the sample sizes based on which the correlations were computed.

`data` data frame containing the variables specified via the formula (and the sample sizes).

`rtoz` logical to specify whether to transform the correlations via Fisher's r-to-z transformation (the default is `FALSE`).

`nfun` a character string to specify how the 'common' sample size within each study should be computed. Possible options are `"min"` (for the minimum), `"harmonic"` (for the harmonic mean), or `"mean"` (for the arithmetic mean). Can also be a function. See 'Details'.

`sparse` logical to specify whether the variance-covariance matrix should be returned as a sparse matrix (the default is `FALSE`).

`...` other arguments.

Details

A meta-analysis of correlation coefficients may involve multiple correlation coefficients extracted from the same study. When these correlations are computed based on the same sample of subjects, then they are typically not independent. The `rcalc` function can be used to create a dataset with the correlation coefficients (possibly transformed with Fisher's r-to-z transformation) and the corresponding variance-covariance matrix. The dataset and variance-covariance matrix can then be further meta-analyzed using the `rma.mv` function.

When computing the covariance between two correlation coefficients, we can distinguish two cases. In the first case, one of the variables involved in the two correlation coefficients is the same. For example, in r_{12} and r_{13}, variable 1 is common to both correlation coefficients. This is sometimes called the (partially) 'overlapping' case. The covariance between the two correlation coefficients, $\text{Cov}[r_{12}, r_{13}]$, then depends on the degree of correlation between variables 2 and 3 (i.e., r_{23}).

In the second case, none of the variables are common to both correlation coefficients. For example, this would be the case if we have correlations r_{12} and r_{34} based on 4 variables. This sometimes called the 'non-overlapping' case. The covariance between the two correlation coefficients, $\text{Cov}[r_{12}, r_{34}]$, then depends on r_{13}, r_{14}, r_{23}, and r_{24}.

Equations for these covariances can be found, for example, in Steiger (1980) and Olkin and Finn (1990).

To use the `rcalc` function, one needs to construct a data frame that contains a study identifier (say `study`), two variable identifiers (say `var1` and `var2`), the corresponding correlation coefficient (say `ri`), and the sample size based on which the correlation coefficient was computed (say `ni`). Then the first argument should be a formula of the form `ri ~ var1 + var2 | study`, argument `ni` is set equal to the variable name containing the sample sizes, and the data frame containing these variables is specified via the `data` argument. When using the function for a single study, one can leave out the study identifier from the formula.

When argument `rtoz` is set to `TRUE`, then the correlations are transformed with Fisher's r-to-z transformation and the variance-covariance matrix is computed for the transformed values.

In some cases, the sample size may not be identical within a study (e.g., r_{12} may have been computed based on 120 subjects while r_{13} was computed based on 118 subjects due to 2 missing values in variable 3). For constructing the variance-covariance matrix, we need to assume a 'common' sample size for all correlation coefficients within the study. Argument `nfun` provides some options for how the common sample size should be computed. Possible options are `"min"` (for using the minimum sample size within a study as the common sample size), `"harmonic"` (for using the harmonic mean), or `"mean"` (for using the arithmetic mean). The default is `"min"`, which is a conservative choice (i.e., it will overestimate the sampling variances of coefficients that were computed based on a sample size that was actually larger than the minimum sample size). One can also specify a function via the `nfun` argument (which should take a numeric vector as input and return a single value).

Value

A list containing the following components:

`dat`	a data frame with the study identifier, the two variable identifiers, a variable pair identifier, the correlation coefficients (possibly transformed with Fisher's r-to-z transformation), and the (common) sample sizes.
`V`	corresponding variance-covariance matrix (given as a sparse matrix when `sparse=TRUE`).

Note that a particular covariance can only be computed when all of the correlation coefficients involved in the covariance equation are included in the dataset. If one or more coefficients needed for the computation are missing, then the resulting covariance will also be missing (i.e., `NA`).

Note

For raw correlation coefficients, the variance-covariance matrix is computed with $n - 1$ in the denominator (instead of n as suggested in Steiger, 1980, and Olkin & Finn, 1990). This is more consistent with the usual equation for computing the sampling variance of a correlation coefficient (with $n - 1$ in the denominator).

For raw and r-to-z transformed coefficients, the variance-covariance matrix will only be computed when the (common) sample size for a study is at least 5.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Olkin, I., & Finn, J. D. (1990). Testing correlated correlations. *Psychological Bulletin*, **108**(2), 330–333. https://doi.org/10.1037/0033-2909.108.2.330

Steiger, J. H. (1980). Tests for comparing elements of a correlation matrix. *Psychological Bulletin*, **87**(2), 245–251. https://doi.org/10.1037/0033-2909.87.2.245

See Also

`rma.mv`, `dat.craft2003`

Examples

```
### copy data into 'dat'
dat <- dat.craft2003

### construct dataset and var-cov matrix of the correlations
tmp <- rcalc(ri ~ var1 + var2 | study, ni=ni, data=dat)
V <- tmp$V
dat <- tmp$dat

### examine data for study 1
dat[dat$study == 1,]
V[dat$study == 1, dat$study == 1]

### examine data for study 6
dat[dat$study == 6,]
V[dat$study == 6, dat$study == 6]

### examine data for study 17
dat[dat$study == 17,]
V[dat$study == 17, dat$study == 17]
```

---

regplot *Scatter Plots / Bubble Plots*

Description

Function to create scatter plots / bubble plots based on meta-regression models.

Usage

```
regplot(x, ...)

## S3 method for class 'rma'
regplot(x, mod, pred=TRUE, ci=TRUE, pi=FALSE, shade=TRUE,
        xlim, ylim, predlim, olim, xlab, ylab, at, digits=2L,
        transf, atransf, targs, level=x$level,
        pch=21, psize, plim=c(0.5,3), col="black", bg="darkgray",
        grid=FALSE, refline, label=FALSE, offset=c(1,1), labsize=1,
        lcol, lwd, lty, legend=FALSE, xvals, ...)

## S3 method for class 'regplot'
points(x, ...)
```

Arguments

x an object of class "rma.uni", "rma.mv", or "rma.glmm" including one or multiple moderators (or an object of class "regplot" for points).

`mod` either a scalar to specify the position of the moderator variable in the model or a character string to specify the name of the moderator variable.

`pred` logical to indicate whether the (marginal) regression line based on the moderator should be added to the plot (the default is `TRUE`). Can also be an object from `predict.rma`. See 'Details'.

`ci` logical to indicate whether the corresponding confidence interval bounds should be added to the plot (the default is `TRUE`).

`pi` logical to indicate whether the corresponding prediction interval bounds should be added to the plot (the default is `FALSE`).

`shade` logical to indicate whether the confidence/prediction interval regions should be shaded (the default is `TRUE`). Can also be a two-element character vector to specify the colors for shading the confidence and prediction interval regions (if shading only the former, a single color can also be specified).

`xlim` x-axis limits. If unspecified, the function tries to set the x-axis limits to some sensible values.

`ylim` y-axis limits. If unspecified, the function tries to set the y-axis limits to some sensible values.

`predlim` optional argument to specify the limits of the (marginal) regression line. If unspecified, the limits are based on the range of the moderator variable.

`olim` optional argument to specify observation/outcome limits. If unspecified, no limits are used.

`xlab` title for the x-axis. If unspecified, the function tries to set an appropriate axis title.

`ylab` title for the y-axis. If unspecified, the function tries to set an appropriate axis title.

`at` position of the y-axis tick marks and corresponding labels. If unspecified, the function tries to set the tick mark positions/labels to some sensible values.

`digits` integer to specify the number of decimal places to which the tick mark labels of the y-axis should be rounded. When specifying an integer (e.g., `2L`), trailing zeros after the decimal mark are dropped for the y-axis labels. When specifying a numeric value (e.g., `2`), trailing zeros are retained.

`transf` optional argument to specify a function that should be used to transform the observed outcomes, predicted values, and confidence/prediction interval bounds (e.g., `transf=exp`; see also transf). If unspecified, no transformation is used.

`atransf` optional argument to specify a function that should be used to transform the y-axis labels (e.g., `atransf=exp`; see also transf). If unspecified, no transformation is used.

`targs` optional arguments needed by the function specified via `transf` or `atransf`.

`level` numeric value between 0 and 100 to specify the confidence/prediction interval level (the default is to take the value from the object).

`pch` plotting symbol to use for the observed outcomes. By default, a filled circle is used. Can also be a vector of values. See `points` for other options.

`psize` optional numeric value to specify the point sizes for the observed outcomes. If unspecified, the point sizes are a function of the model weights. Can also be a vector of values. Can also be a character string (either `"seinv"` or `"vinv"`) to make the point sizes proportional to the inverse standard errors or inverse sampling variances.

`plim` numeric vector of length 2 to scale the point sizes (ignored when a numeric value or vector is specified for `psize`). See ‘Details’.

`col` character string to specify the name of a color to use for plotting the observed outcomes (the default is `"black"`). Can also be a vector of color names.

`bg` character string to specify the name of a background color for open plot symbols (the default is `"darkgray"`). Can also be a vector of color names.

`grid` logical to specify whether a grid should be added to the plot. Can also be a color name.

`refline` optional numeric value to specify the location of a horizontal reference line that should be added to the plot.

`label` argument to control the labeling of the points (the default is `FALSE`). See ‘Details’.

`offset` argument to control the distance between the points and the corresponding labels. See ‘Details’.

`labsize` numeric value to control the size of the labels.

`lcol` optional vector of (up to) four elements to specify the color of the regression line, of the confidence interval bounds, of the prediction interval bounds, and of the horizontal reference line.

`lty` optional vector of (up to) four elements to specify the line type of the regression line, of the confidence interval bounds, of the prediction interval bounds, and of the horizontal reference line.

`lwd` optional vector of (up to) four elements to specify the line width of the regression line, of the confidence interval bounds, of the prediction interval bounds, and of the horizontal reference line.

`legend` logical to indicate whether a legend should be added to the plot (the default is `FALSE`). Can also be a keyword to indicate the position of the legend (see `legend`).

`xvals` optional numeric vector to specify the values of the moderator for which predicted values should be computed. Needs to be specified when passing an object from `predict.rma` to the `pred` argument. See ‘Details’.

`...` other arguments.

Details

The function draws a scatter plot of the values of a moderator variable in a meta-regression model (on the x-axis) against the observed effect sizes or outcomes (on the y-axis). The regression line from the model (with corresponding confidence interval bounds) is added to the plot by default. These types of plots are also often referred to as ‘bubble plots’ as the points are typically drawn in different sizes to reflect their precision or weight in the model.

By default (i.e., when psize is not specified), the size of the points is a function of the square root of the model weights. This way, their area is proportional to the the weights. However, the point sizes are rescaled so that the smallest point size is plim[1] and the largest point size is plim[2]. As a result, their relative sizes (i.e., areas) no longer exactly correspond to their relative weights. If exactly relative point sizes are desired, one can set plim[2] to NA, in which case the points are rescaled so that the smallest point size corresponds to plim[1] and all other points are scaled accordingly. As a result, the largest point may be very large. Alternatively, one can set plim[1] to NA, in which case the points are rescaled so that the largest point size corresponds to plim[2] and all other points are scaled accordingly. As a result, the smallest point may be very small. To avoid the latter, one can also set plim[3], which enforces a minimal point size.

One can also set psize to a scalar (e.g., psize=1) to avoid that the points are drawn in different sizes. One can also specify the point sizes manually by passing a vector of the appropriate length to psize. Finally, one can also set psize to either "seinv" or "vinv" to make the point sizes proportional to the inverse standard errors or inverse sampling variances.

For a model with more than one predictor, the regression line reflects the ‘marginal’ relationship between the chosen moderator and the effect sizes or outcomes (i.e., all other moderators except the one being plotted are held constant at their means).

With the label argument, one can control whether points in the plot will be labeled. If label="all" (or label=TRUE), all points in the plot will be labeled. If label="ciout" or label="piout", points falling outside of the confidence/prediction interval will be labeled. Alternatively, one can set this argument to a logical or numeric vector to specify which points should be labeled. The labels are placed above the points when they fall above the regression line and otherwise below. With the offset argument, one can adjust the distance between the labels and the corresponding points. This can either be a single numeric value, which is used as a multiplicative factor for the point sizes (so that the distance between labels and points is larger for larger points) or a numeric vector with two values, where the first is used as an additive factor independent of the point sizes and the second again as a multiplicative factor for the point sizes. The values are given as percentages of the y-axis range. It may take some trial and error to find two values for the offset argument so that the labels are placed right next to the boundary of the points. With labsize, one can control the size of the labels.

One can also pass an object from predict.rma to the pred argument. This can be useful when the meta-regression model reflects a more complex relationship between the moderator variable and the effect sizes or outcomes (e.g., when using polynomials or splines) or when the model involves interactions. In this case, one also needs to specify the xvals argument. See ‘Examples’.

Value

An object of class "regplot" with components:

slab	the study labels
ids	the study ids
xi	the x-axis coordinates of the points that were plotted.
yi	the y-axis coordinates of the points that were plotted.
pch	the plotting symbols of the points that were plotted.
psize	the point sizes of the points that were plotted.
col	the colors of the points that were plotted.

bg	the background colors of the points that were plotted.
label	logical vector indicating whether a point was labeled or not.

Note that the object is returned invisibly. Using points.regplot, one can redraw the points in case one wants to superimpose the points on top of any elements that were added manually to the plot (see 'Examples').

Note

For certain types of models, it may not be possible to draw the prediction interval bounds (if this is the case, a warning will be issued).

When specifying vectors for pch, psize, col, bg, and/or label, the variables specified are assumed to be of the same length as the data passed to the model fitting function. Any subsetting and removal of studies with missing values is automatically applied to the variables specified via these arguments.

If the outcome measure used for creating the plot is bounded (e.g., correlations are bounded between -1 and +1, proportions are bounded between 0 and 1), one can use the olim argument to enforce those limits (the observed outcomes and confidence/prediction intervals cannot exceed those bounds then).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Thompson, S. G., & Higgins, J. P. T. (2002). How should meta-regression analyses be undertaken and interpreted? *Statistics in Medicine*, **21**(11), 1559–1573. https://doi.org/10.1002/sim.1187

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.glmm, rma.mv

Examples

```
### copy BCG vaccine data into 'dat'
dat <- dat.bcg

### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat)

############################################################################

### fit mixed-effects model with absolute latitude as a moderator
res <- rma(yi, vi, mods = ~ ablat, data=dat)
res

### draw plot
```

```
regplot(res, mod="ablat", xlab="Absolute Latitude")

### adjust x-axis limits and back-transform to risk ratios
regplot(res, mod="ablat", xlab="Absolute Latitude", xlim=c(0,60), transf=exp)

### also extend the prediction limits for the regression line
regplot(res, mod="ablat", xlab="Absolute Latitude", xlim=c(0,60), predlim=c(0,60), transf=exp)

### add the prediction interval to the plot, add a reference line at 1, and add a legend
regplot(res, mod="ablat", pi=TRUE, xlab="Absolute Latitude",
        xlim=c(0,60), predlim=c(0,60), transf=exp, refline=1, legend=TRUE)

### label points outside of the prediction interval
regplot(res, mod="ablat", pi=TRUE, xlab="Absolute Latitude",
        xlim=c(0,60), predlim=c(0,60), transf=exp, refline=1, legend=TRUE,
        label="piout", labsize=0.8)

############################################################################

### fit mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)
res

### plot the marginal relationships
regplot(res, mod="ablat", xlab="Absolute Latitude")
regplot(res, mod="year",  xlab="Publication Year")

############################################################################

### fit a quadratic polynomial meta-regression model
res <- rma(yi, vi, mods = ~ ablat + I(ablat^2), data=dat)
res

### compute predicted values using predict()
xs <- seq(0,60,length=601)
tmp <- predict(res, newmods=cbind(xs, xs^2))

### can now pass these results to the 'pred' argument (and have to specify xvals accordingly)
regplot(res, mod="ablat", pred=tmp, xlab="Absolute Latitude", xlim=c(0,60), xvals=xs)

### back-transform to risk ratios and add reference line
regplot(res, mod="ablat", pred=tmp, xlab="Absolute Latitude", xlim=c(0,60), xvals=xs,
        transf=exp, refline=1)

############################################################################

### fit a model with an interaction between a quantitative and a categorical predictor
### (note: just for illustration purposes; this model is too complex for this dataset)
res <- rma(yi, vi, mods = ~ ablat * alloc, data=dat)
res

### draw bubble plot but do not add regression line or CI
tmp <- regplot(res, mod="ablat", xlab="Absolute Latitude", xlim=c(0,60), pred=FALSE, ci=FALSE)
```

```
### add regression lines for the three alloc levels
xs <- seq(0, 60, length=100)
preds <- predict(res, newmods=cbind(xs, 0, 0, 0, 0))
lines(xs, preds$pred, lwd=3)
preds <- predict(res, newmods=cbind(xs, 1, 0, xs, 0))
lines(xs, preds$pred, lwd=3)
preds <- predict(res, newmods=cbind(xs, 0, 1, 0, xs))
lines(xs, preds$pred, lwd=3)

### add points back to the plot (so they are on top of the lines)
points(tmp)
```

regtest *Regression Test for Funnel Plot Asymmetry*

Description

The function can be used to carry out (various versions of) Egger's regression test for funnel plot asymmetry.

Usage

```
regtest(x, ...)

## S3 method for class 'rma'
regtest(x, model="rma", predictor="sei", ret.fit=FALSE, digits, ...)

## Default S3 method:
regtest(x, vi, sei, ni, subset, model="rma", predictor="sei", ret.fit=FALSE, digits, ...)
```

Arguments

`x` an object of class `"rma"` or a vector with the observed effect sizes or outcomes.

`vi` vector with the corresponding sampling variances (needed if `x` is a vector with the observed effect sizes or outcomes).

`sei` vector with the corresponding standard errors (note: only one of the two, `vi` or `sei`, needs to be specified).

`ni` vector with the corresponding sample sizes.

`subset` optional (logical or numeric) vector to specify the subset of studies that should be included in the test. Only relevant when passing a vector via `x`.

`model` either `"rma"` or `"lm"` to indicate the type of model to use for the regression test. See 'Details'.

`predictor` either `"sei"` `"vi"`, `"ni"`, `"ninv"`, `"sqrtni"`, or `"sqrtninv"` to indicate the predictor to use for the regression test. See 'Details'.

`ret.fit`	logical to specify whether the full results from the fitted model should also be returned.
`digits`	integer to specify the number of decimal places to which the printed results should be rounded (the default is to take the value from the object or is 4 for the default method).
`...`	other arguments.

Details

Various tests for funnel plot asymmetry have been suggested in the literature, including the rank correlation test by Begg and Mazumdar (1994) and the regression test by Egger et al. (1997). Extensions, modifications, and further developments of the regression test are described (among others) by Macaskill, Walter, and Irwig (2001), Sterne and Egger (2005), Harbord, Egger, and Sterne (2006), Peters et al. (2006), Rücker et al. (2008), and Moreno et al. (2009). The various versions of the regression test differ in terms of the model (either a weighted regression model with a multiplicative dispersion term or a fixed/random/mixed-effects meta-regression model is used), in terms of the predictor variable that the observed effect sizes or outcomes are hypothesized to be related to when publication bias is present (suggested predictors include the standard error, the sampling variance, and the sample size or transformations thereof), and in terms of the outcome measure used (e.g., for 2×2 table data, one has the choice between various outcome measures). The idea behind the various tests is the same though: If there is a relationship between the observed effect sizes or outcomes and the chosen predictor, then this usually implies asymmetry in the funnel plot, which in turn may be an indication of publication bias.

The `regtest` function can be used to carry out various versions of the regression test. The model is chosen via the `model` argument, with `model="lm"` for weighted regression with a multiplicative dispersion term or `model="rma"` for the meta-analytic models. In the latter case, arguments such as `method`, `weighted`, and `test` used during the initial model fitting are also used for the regression test. Therefore, if one wants to conduct the regression test with a random/mixed-effects model, one should first fit a random-effects model with the `rma` function and then use the `regtest` function on the fitted model object.

The predictor is chosen via the `predictor` argument:

- `predictor="sei"` for the standard error,
- `predictor="vi"` for the sampling variance,
- `predictor="ni"` for the sample size,
- `predictor="ninv"` for the inverse of the sample size,
- `predictor="sqrtni"` for the square root transformed sample size, and
- `predictor="sqrtninv"` for the inverse of the square root transformed sample size.

For predictors based on the sample size, the object `x` obviously must contain the information about the sample sizes. This will automatically be the case when `measure` was *not* equal to `"GEN"` or the `ni` values were explicitly specified during the initial model fitting.

If the model passed to the `regtest` function already included one or more moderators, then `regtest` will add the chosen predictor to the moderator(s) already included in the model. This way, one can test for funnel plot asymmetry after accounting first for the influence of the moderator(s) already included.

One can also pass a vector with the observed effect sizes or outcomes (via `x`) and the corresponding sampling variances via `vi` (or the standard errors via `sei`) directly to the function (in this case, the `regtest.default` function is used). When the predictor is the sample size or a transformation thereof, then `ni` needs to be specified here as well.

The outcome measure used for the regression test is simply determined by what measure was used in fitting the original model (or what values are passed to `regtest.default`).

The model used for conducting the regression test can also be used to obtain a 'limit estimate' of the (average) true effect or outcome. In particular, when the standard error, sampling variance, or inverse (square root) sample size is used as the predictor, the model intercept in essence reflects the estimate under infinite precision. This is sometimes (cautiously) interpreted as an estimate of the (average) true effect or outcome that is adjusted for publication bias.

Value

An object of class `"regtest"`. The object is a list containing the following components:

`model`	the model used for the regression test.
`predictor`	the predictor used for the regression test.
`zval`	the value of the test statistic.
`pval`	the corresponding p-value
`dfs`	the degrees of freedom of the test statistic (if the test is based on a t-distribution).
`fit`	the full results from the fitted model.
`est`	the limit estimate (only for predictors `"sei"` `"vi"`, `"ninv"`, or `"sqrtninv"` and when the model does not contain any additional moderators; `NULL` otherwise)
`ci.lb`	lower bound of the confidence interval for the limit estimate.
`ci.ub`	upper bound of the confidence intervals for the limit estimate.

The results are formatted and printed with the `print.regtest` function.

Note

The classical 'Egger test' is obtained by setting `model="lm"` and `predictor="sei"`. For the random/mixed-effects version of the Egger test, one should first fit a random-effects model to the data and then set `model="rma"` and `predictor="sei"` when using the `regtest` function. See Sterne and Egger (2005) for details on these two types of models/tests.

When conducting a classical 'Egger test', the test of the limit estimate is the same as the 'precision-effect test' (PET) of Stanley and Doucouliagos (2014). The limit estimate when using the sampling variance as predictor is sometimes called the 'precision-effect estimate with SE' (PEESE) (Stanley & Doucouliagos, 2014). A conditional procedure where we use the limit estimate when PET is not significant (i.e., when using the standard error as predictor) and the PEESE (i.e., when using the sampling variance as predictor) when PET is significant is sometimes called the PET-PEESE procedure (Stanley & Doucouliagos, 2014).

All of the tests do not directly test for publication bias, but for a relationship between the observed effect sizes or outcomes and the chosen predictor. If such a relationship is present, then this usually implies asymmetry in the funnel plot, which in turn may be an indication of publication bias. However, it is important to keep in mind that there can be other reasons besides publication bias that could lead to asymmetry in the funnel plot.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Begg, C. B., & Mazumdar, M. (1994). Operating characteristics of a rank correlation test for publication bias. *Biometrics*, **50**(4), 1088–1101. https://doi.org/10.2307/2533446

Egger, M., Davey Smith, G., Schneider, M., & Minder, C. (1997). Bias in meta-analysis detected by a simple, graphical test. *British Medical Journal*, **315**(7109), 629–634. https://doi.org/10.1136/bmj.315.7109.629

Harbord, R. M., Egger, M., & Sterne, J. A. C. (2006). A modified test for small-study effects in meta-analyses of controlled trials with binary endpoints. *Statistics in Medicine*, **25**(20), 3443–3457. https://doi.org/10.1002/sim.2380

Macaskill, P., Walter, S. D., & Irwig, L. (2001). A comparison of methods to detect publication bias in meta-analysis. *Statistics in Medicine*, **20**(4), 641–654. https://doi.org/10.1002/sim.698

Moreno, S. G., Sutton, A. J., Ades, A. E., Stanley, T. D., Abrams, K. R., Peters, J. L., & Cooper, N. J. (2009). Assessment of regression-based methods to adjust for publication bias through a comprehensive simulation study. *BMC Medical Research Methodology*, **9**, 2. https://doi.org/10.1186/1471-2288-9-2

Peters, J. L., Sutton, A. J., Jones, D. R., Abrams, K. R., & Rushton, L. (2006). Comparison of two methods to detect publication bias in meta-analysis. *Journal of the American Medical Association*, **295**(6), 676–680. https://doi.org/10.1001/jama.295.6.676

Rücker, G., Schwarzer, G., & Carpenter, J. (2008). Arcsine test for publication bias in meta-analyses with binary outcomes. *Statistics in Medicine*, **27**(5), 746–763. https://doi.org/10.1002/sim.2971

Stanley, T. D., & Doucouliagos, H. (2014). Meta-regression approximations to reduce publication selection bias. *Research Synthesis Methods*, **5**(1), 60–78. https://doi.org/10.1002/jrsm.1095

Sterne, J. A. C., & Egger, M. (2005). Regression methods to detect publication and other bias in meta-analysis. In H. R. Rothstein, A. J. Sutton, & M. Borenstein (Eds.) *Publication bias in meta-analysis: Prevention, assessment, and adjustments* (pp. 99–110). Chichester, England: Wiley.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

ranktest

Examples

```
### copy data into 'dat' and examine data
dat <- dat.egger2001

### calculate log odds ratios and corresponding sampling variances (but remove ISIS-4 trial)
dat <- escalc(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, subset=-16)

### fit random-effects model
res <- rma(yi, vi, data=dat)
res
```

```
### classical Egger test
regtest(res, model=”lm”)

### random/mixed-effects version of the Egger test
regtest(res)

### same tests, but passing outcomes directly
regtest(dat$yi, dat$vi, model=”lm”)
regtest(dat$yi, dat$vi)

### examples using the sample size (or a transformation thereof) as predictor
regtest(res, model=”lm”, predictor=”ni”)
regtest(res, model=”lm”, predictor=”ninv”)
regtest(res, model=”rma”, predictor=”ni”)
regtest(res, model=”rma”, predictor=”ninv”)

### if dat$yi is computed with escalc(), sample size information is stored in attributes
dat$yi

### then this will work
regtest(dat$yi, dat$vi, predictor=”ni”)

### otherwise have to supply sample sizes manually
dat$ni <- with(dat, n1i + n2i)
dat$yi <- c(dat$yi) # this removes the 'ni' attribute from 'yi'
regtest(dat$yi, dat$vi, ni=dat$ni, predictor=”ni”)

### standard funnel plot (with standard error on y-axis)
funnel(res, refline=0)

### regression test (by default the standard error is used as predictor)
reg <- regtest(res)
reg

### add regression line to funnel plot
se <- seq(0,1.8,length=100)
lines(coef(reg$fit)[1] + coef(reg$fit)[2]*se, se, lwd=2)

### regression test (using the sampling variance as predictor)
reg <- regtest(res, predictor=”vi”)

### add regression line to funnel plot (using the sampling variance as predictor)
lines(coef(reg$fit)[1] + coef(reg$fit)[2]*se^2, se, lwd=2)

### testing for asymmetry after accounting for the influence of a moderator
res <- rma(yi, vi, mods = ~ year, data=dat)
regtest(res, model=”lm”)
regtest(res)
```

replmiss *Replace Missing Values in a Vector*

Description

Function to replace missing (NA) values in a vector.

Usage

```
replmiss(x, y)
```

Arguments

x	vector that may include one or more missing values.
y	either a scalar or a vector of the same length as x with the value(s) to replace missing values with.

Value

Vector x with the missing values replaced based on the scalar or vector y.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

Examples

```
x <- c(4,2,7,NA,1,NA,5)
x <- replmiss(x,0)
x

x <- c(4,2,7,NA,1,NA,5)
y <- c(2,3,6,5,8,1,2)
x <- replmiss(x,y)
x
```

---

reporter *Dynamically Generated Analysis Reports for 'rma.uni' Objects*

Description

The function dynamically generates analysis reports for objects of class "rma.uni".

Usage

```
reporter(x, ...)

## S3 method for class 'rma.uni'
reporter(x, dir, filename, format="html_document", open=TRUE,
         digits, forest, funnel, footnotes=FALSE, verbose=TRUE, ...)
```

Arguments

`x` an object of class `"rma.uni"`.

`dir` optional character string to specify the directory for creating the report. If unspecified, `tempdir` will be used.

`filename` optional character string to specify the filename (without file extension) for the report. If unspecified, the function sets a filename automatically.

`format` output format for the report (either `html_document`, `pdf_document`, or `word_document`). Can be abbreviated. See ‘Note’.

`open` logical to specify whether the report should be opened after it has been generated (the default is `TRUE`). See ‘Note’.

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.

`forest` either a logical which will suppress the drawing of the forest plot when set to `FALSE` or a character string with arguments to be added to the call to `forest.rma` for generating the forest plot.

`funnel` either a logical which will suppress the drawing of the funnel plot when set to `FALSE` or a character string with arguments to be added to the call to `funnel.rma` for generating the funnel plot.

`footnotes` logical to specify whether additional explanatory footnotes should be added to the report (the default is `FALSE`).

`verbose` logical to specify whether information on the progress of the report generation should be provided (the default is `TRUE`).

`...` other arguments.

Details

The function dynamically generates an analysis report based on the model object. The report includes information about the model that was fitted, the distribution of the observed effect sizes or outcomes, the estimate of the average outcome based on the fitted model, tests and statistics that are informative about potential (residual) heterogeneity in the outcomes, checks for outliers and/or influential studies, and tests for funnel plot asymmetry. By default, a forest plot and a funnel plot are also provided (these can be suppressed by setting `forest=FALSE` and/or `funnel=FALSE`).

Value

The function generates either a html, pdf, or docx file and returns (invisibly) the path to the generated document.

Note

Since the report is created based on an R markdown document that is generated by the function, the rmarkdown package and pandoc must be installed.

To render the report into a pdf document (i.e., using `format="pdf_document"`) requires a LaTeX installation. If LaTeX is not already installed, you could try using the tinytex package to install a lightweight LaTeX distribution based on TeX Live.

Once the report is generated, the function tries to open the output file (either a .html, .pdf, or .docx file) with an appropriate application (if open=TRUE). This will only work when an appropriate application for the file type is installed and associated with the extension.

If filename is unspecified, the default is to use report, followed by an underscore (i.e., _) and the name of the object passed to the function. Both the R markdown file (with extension .rmd) and the actual report (with extension .html, .pdf, or .docx) are named accordingly. To generate the report, the model object is also saved to a file (with the same filename as above, but with extension .rdata). Also, files references.bib and apa.csl are copied to the same directory (these files are needed to generate the references in APA format).

Since the report is put together based on predefined text blocks, the writing is not very elegant. Also, using personal pronouns (‘I’ or ‘we’) does not make sense for such a report, so a lot of passive voice is used.

The generated report provides an illustration of how the results of the model can be reported, but is not a substitute for a careful examination of the results.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni

Examples

```
### copy BCG vaccine data into 'dat'
dat <- dat.bcg

### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat,
              slab=paste(author, ", ", year, sep=""))

### fit random-effects model
res <- rma(yi, vi, data=dat)

## Not run:
### generate pdf report
reporter(res)
## End(Not run)
```

---

residuals.rma *Residual Values based on 'rma' Objects*

Description

The residuals, rstandard, and rstudent functions compute residuals, corresponding standard errors, and standardized residuals for models fitted with the rma.uni, rma.mh, rma.peto, and rma.mv functions.

Usage

```
## S3 method for class 'rma'
residuals(object, type="response", ...)

## S3 method for class 'rma.uni'
rstandard(model, digits, type="marginal", ...)
## S3 method for class 'rma.mh'
rstandard(model, digits, ...)
## S3 method for class 'rma.peto'
rstandard(model, digits, ...)
## S3 method for class 'rma.mv'
rstandard(model, digits, cluster, ...)

## S3 method for class 'rma.uni'
rstudent(model, digits, progbar=FALSE, ...)
## S3 method for class 'rma.mh'
rstudent(model, digits, progbar=FALSE, ...)
## S3 method for class 'rma.peto'
rstudent(model, digits, progbar=FALSE, ...)
## S3 method for class 'rma.mv'
rstudent(model, digits, progbar=FALSE, cluster,
         reestimate=TRUE, parallel="no", ncpus=1, cl=NULL, ...)
```

Arguments

object	an object of class "rma" (for residuals).
type	the type of residuals which should be returned. For residuals, the alternatives are: "response" (default), "rstandard", "rstudent", and "pearson". For rstandard.rma.uni, the alternatives are: "marginal" (default) and "conditional". See 'Details'.
model	an object of class "rma" (for residuals) or an object of class "rma.uni", "rma.mh", "rma.peto", or "rma.mv" (for rstandard and rstudent).
cluster	optional vector to specify a clustering variable to use for computing cluster-level multivariate standardized residuals (only for "rma.mv" objects).

`reestimate` logical to specify whether variance/correlation components should be re-estimated after deletion of the ith case when computing externally standardized residuals for `"rma.mv"` objects (the default is `TRUE`).

`parallel` character string to specify whether parallel processing should be used (the default is `"no"`). For parallel processing, set to either `"snow"` or `"multicore"`. See ‘Details’.

`ncpus` integer to specify the number of processes to use in the parallel processing.

`cl` optional cluster to use if `parallel="snow"`. If not supplied, a cluster on the local machine is created for the duration of the call.

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.

`progbar` logical to specify whether a progress bar should be shown (only for `rstudent`) (the default is `FALSE`).

`...` other arguments.

Details

The observed residuals (obtained with `residuals`) are simply equal to the ‘observed - fitted’ values. These can be obtained with `residuals(object)` (using the default `type="response"`).

Dividing the observed residuals by the model-implied standard errors of the observed effect sizes or outcomes yields Pearson (or semi-standardized) residuals. These can be obtained with `residuals(object,type="pearson"`

Dividing the observed residuals by their corresponding standard errors yields (internally) standardized residuals. These can be obtained with `rstandard(model)` or `residuals(object,type="rstandard")`.

With `rstudent(model)` (or `residuals(object,type="rstudent")`), one can obtain the externally standardized residuals (also called standardized deleted residuals or (externally) studentized residuals). The externally standardized residual for the ith case is obtained by deleting the ith case from the dataset, fitting the model based on the remaining cases, calculating the predicted value for the ith case based on the fitted model, taking the difference between the observed and the predicted value for the ith case (which yields the deleted residual), and then standardizing the deleted residual based on its standard error.

If a particular case fits the model, its standardized residual follows (asymptotically) a standard normal distribution. A large standardized residual for a case therefore may suggest that the case does not fit the assumed model (i.e., it may be an outlier).

For `"rma.uni"` objects, `rstandard(model,type="conditional")` computes conditional residuals, which are the deviations of the observed effect sizes or outcomes from the best linear unbiased predictions (BLUPs) of the study-specific true effect sizes or outcomes (see `blup.rma.uni`).

For `"rma.mv"` objects, one can specify a clustering variable (via the `cluster` argument). If specified, `rstandard(model)` and `rstudent(model)` also compute cluster-level multivariate (internally or externally) standardized residuals. If all outcomes within a cluster fit the model, then the multivariate standardized residual for the cluster follows (asymptotically) a chi-square distribution with k_i degrees of freedom (where k_i denotes the number of outcomes within the cluster).

See also `influence.rma.uni` and `influence.rma.mv` for other leave-one-out diagnostics that are useful for detecting influential cases in models fitted with the `rma.uni` and `rma.mv` functions.

Value

Either a vector with the residuals of the requested type (for `residuals`) or an object of class `"list.rma"`, which is a list containing the following components:

`resid`	observed residuals (for `rstandard`) or deleted residuals (for `rstudent`).
`se`	corresponding standard errors.
`z`	standardized residuals (internally standardized for `rstandard` or externally standardized for `rstudent`).

When a clustering variable is specified for `"rma.mv"` objects, the returned object is a list with the first element (named `obs`) as described above and a second element (named `cluster` of class `"list.rma"` with:

`X2`	cluster-level multivariate standardized residuals.
`k`	number of observed effect sizes or outcomes within the clusters.

The object is formatted and printed with `print.list.rma`.

Note

Right now, the externally standardized residuals (obtained with `rstudent`) are calculated by refitting the model k times (where k is the number of cases). Depending on how large k is, it may take a few moments to finish the calculations. For complex models fitted with `rma.mv`, this can become computationally expensive.

On machines with multiple cores, one can usually speed things up by delegating the model fitting to separate worker processes, that is, by setting `parallel="snow"` or `parallel="multicore"` and `ncpus` to some value larger than 1 (only for objects of class `"rma.mv"`). Parallel processing makes use of the `parallel` package, using the `makePSOCKcluster` and `parLapply` functions when `parallel="snow"` or using `mclapply` when `parallel="multicore"` (the latter only works on Unix/Linux-alikes). With `parallel::detectCores()`, one can check on the number of available cores on the local machine.

Alternatively (or in addition to using parallel processing), one can also set `reestimate=FALSE`, in which case any variance/correlation components in the model are not re-estimated after deleting the ith case from the dataset. Doing so only yields an approximation to the externally standardized residuals (and the cluster-level multivariate standardized residuals) that ignores the influence of the ith case on the variance/correlation components, but is considerably faster (and often yields similar results).

It may not be possible to fit the model after deletion of the ith case from the dataset. This will result in `NA` values for that case when calling `rstudent`.

Also, for `"rma.mv"` objects with a clustering variable specified, it may not be possible to compute the cluster-level multivariate standardized residual for a particular cluster (if the var-cov matrix of the residuals within a cluster is not of full rank). This will result in `NA` for that cluster.

For objects of class `"rma.mh"` and `"rma.peto"`, `rstandard` actually computes Pearson (or semi-standardized) residuals.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. San Diego, CA: Academic Press.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Viechtbauer, W., & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**(2), 112–125. https://doi.org/10.1002/jrsm.11

See Also

rma.uni, rma.mh, rma.peto, rma.glmm, rma.mv, influence.rma.uni, influence.rma.mv

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit random-effects model
res <- rma(yi, vi, data=dat)

### compute the studentized residuals
rstudent(res)

### fit mixed-effects model with absolute latitude as moderator
res <- rma(yi, vi, mods = ~ ablat, data=dat)

### compute the studentized residuals
rstudent(res)
```

rma.glmm *Meta-Analysis via Generalized Linear (Mixed-Effects) Models*

Description

Function to fit meta-analytic fixed- and random/mixed-effects models with or without moderators via generalized linear (mixed-effects) models. See below and the documentation of the **metafor-package** for more details on these models.

Usage

```
rma.glmm(ai, bi, ci, di, n1i, n2i, x1i, x2i, t1i, t2i, xi, mi, ti, ni,
         mods, measure, intercept=TRUE, data, slab, subset,
         add=1/2, to="only0", drop00=TRUE, vtype="LS",
         model="UM.FS", method="ML", test="z",
         level=95, digits, btt, nAGQ=7, verbose=FALSE, control, ...)
```

Arguments

`ai` see below and the documentation of the `escalc` function for more details.

`bi` see below and the documentation of the `escalc` function for more details.

`ci` see below and the documentation of the `escalc` function for more details.

`di` see below and the documentation of the `escalc` function for more details.

`n1i` see below and the documentation of the `escalc` function for more details.

`n2i` see below and the documentation of the `escalc` function for more details.

`x1i` see below and the documentation of the `escalc` function for more details.

`x2i` see below and the documentation of the `escalc` function for more details.

`t1i` see below and the documentation of the `escalc` function for more details.

`t2i` see below and the documentation of the `escalc` function for more details.

`xi` see below and the documentation of the `escalc` function for more details.

`mi` see below and the documentation of the `escalc` function for more details.

`ti` see below and the documentation of the `escalc` function for more details.

`ni` see below and the documentation of the `escalc` function for more details.

`mods` optional argument to include one or more moderators in the model. A single moderator can be given as a vector of length k specifying the values of the moderator. Multiple moderators are specified by giving a matrix with k rows and as many columns as there are moderator variables. Alternatively, a model `formula` can be used to specify the model. See 'Details'.

`measure` character string to specify the outcome measure to use for the meta-analysis. Possible options are the odds ratio (`"OR"`), the incidence rate ratio (`"IRR"`), the logit transformed proportion (`"PLO"`), or the log transformed incidence rate (`"IRLN"`).

`intercept` logical to specify whether an intercept should be added to the model (the default is `TRUE`).

`data` optional data frame containing the data supplied to the function.

`slab` optional vector with labels for the k studies.

`subset` optional (logical or numeric) vector to specify the subset of studies that should be used for the analysis.

`add` non-negative number to specify the amount to add to zero cells, counts, or frequencies when calculating the observed effect sizes or outcomes of the individual studies. See below and the documentation of the `escalc` function for more details.

`to` character string to specify when the values under `add` should be added (either `"only0"`, `"all"`, `"if0all"`, or `"none"`). See below and the documentation of the `escalc` function for more details.

`drop00` logical to specify whether studies with no cases/events (or only cases) in both groups should be dropped. See the documentation of the `escalc` function for more details.

`vtype` character string to specify the type of sampling variances to calculate when calculating the observed effect sizes or outcomes. See the documentation of the `escalc` function for more details.

`model` character string to specify the general model type to use for the analysis (either `"UM.FS"` (the default), `"UM.RS"`, `"CM.EL"`, or `"CM.AL"`). See ‘Details’.

`method` character string to specify whether a fixed- or a random/mixed-effects model should be fitted. A fixed-effects model (with or without moderators) is fitted when using `method="FE"`. Random/mixed-effects models are fitted by setting `method="ML"` (the default). See ‘Details’.

`test` character string to specify how test statistics and confidence intervals for the fixed effects should be computed. By default (`test="z"`), Wald-type tests and CIs are obtained, which are based on a standard normal distribution. When `test="t"`, a t-distribution is used instead. See ‘Details’.

`level` numeric value between 0 and 100 to specify the confidence interval level (the default is 95).

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is 4.

`btt` optional vector of indices to specify which coefficients to include in the omnibus test of moderators. Can also be a string to grep for. See ‘Details’.

`nAGQ` positive integer to specify the number of points per axis for evaluating the adaptive Gauss-Hermite approximation to the log-likelihood. The default is 7. Setting this to 1 corresponds to the Laplacian approximation. See ‘Note’.

`verbose` logical to specify whether output should be generated on the progress of the model fitting (the default is `FALSE`). Can also be an integer. Values > 1 generate more verbose output. See ‘Note’.

`control` optional list of control values for the estimation algorithms. If unspecified, default values are defined inside the function. See ‘Note’.

`...` additional arguments.

Details

Specifying the Data

The function can be used in conjunction with the following effect size or outcome measures:

- `measure="OR"` for odds ratios (analyzed in log units)
- `measure="IRR"` for incidence rate ratios (analyzed in log units)
- `measure="PLO"` for logit transformed proportions (i.e., log odds)
- `measure="IRLN"` for log transformed incidence rates.

The `escalc` function describes the data/arguments that should be specified/used for these measures.

Specifying the Model

A variety of model types are available when analyzing 2×2 table data (i.e., when `measure="OR"`) or two-group event count data (i.e., when `measure="IRR"`):

- model="UM.FS" for an unconditional generalized linear mixed-effects model with fixed study effects
- model="UM.RS" for an unconditional generalized linear mixed-effects model with random study effects
- model="CM.AL" for a conditional generalized linear mixed-effects model (approximate likelihood)
- model="CM.EL" for a conditional generalized linear mixed-effects model (exact likelihood).

For measure="OR", models "UM.FS" and "UM.RS" are essentially (mixed-effects) logistic regression models, while for measure="IRR", these models are (mixed-effects) Poisson regression models. A choice must be made on how to model study level variability (i.e., differences in outcomes across studies irrespective of group membership). One can choose between using fixed study effects (which means that k dummy variables are added to the model) or random study effects (which means that random effects corresponding to the levels of the study factor are added to the model).

The conditional model (model="CM.EL") avoids having to model study level variability by conditioning on the total numbers of cases/events in each study. For measure="OR", this leads to a non-central hypergeometric distribution for the data within each study and the corresponding model is then a (mixed-effects) conditional logistic model. Fitting this model can be difficult and computationally expensive. When the number of cases in each study is small relative to the group sizes, one can approximate the exact likelihood by a binomial distribution, which leads to a regular (mixed-effects) logistic regression model (model="CM.AL"). For measure="IRR", the conditional model leads directly to a binomial distribution for the data within each study and the resulting model is again a (mixed-effects) logistic regression model (no approximate likelihood model is needed here).

When analyzing proportions (i.e., measure="PLO") or incidence rates (i.e., measure="IRLN") of individual groups, the model type is always a (mixed-effects) logistic or Poisson regression model, respectively (i.e., the model argument is not relevant here).

Aside from choosing the general model type, one has to decide whether to fit a fixed- or random-effects model to the data. A *fixed-effects model* is fitted by setting method="FE". A *random-effects model* is fitted by setting method="ML" (the default). Note that random-effects models with dichotomous data are often referred to as 'binomial-normal' models in the meta-analytic literature. Analogously, for event count data, such models could be referred to as 'Poisson-normal' models.

One or more moderators can be included in all of these models via the mods argument. A single moderator can be given as a (row or column) vector of length k specifying the values of the moderator. Multiple moderators are specified by giving an appropriate model matrix (i.e., X) with k rows and as many columns as there are moderator variables (e.g., mods = cbind(mod1,mod2,mod3), where mod1, mod2, and mod3 correspond to the names of the variables for three moderator variables). The intercept is added to the model matrix by default unless intercept=FALSE.

Alternatively, one can use standard formula syntax to specify the model. In this case, the mods argument should be set equal to a one-sided formula of the form mods = ~ model (e.g., mods = ~ mod1 + mod2 + mod3). Interactions, polynomial terms, and factors can be easily added to the model in this manner. When specifying a model formula via the mods argument, the intercept argument is ignored. Instead, the inclusion/exclusion of the intercept is controlled by the specified formula (e.g., mods = ~ mod1 + mod2 + mod3 -1 would lead to the removal of the intercept).

Fixed-, Saturated-, and Random/Mixed-Effects Models

When fitting a particular model, actually up to three different models are fitted within the function:

- the fixed-effects model (i.e., where τ^2 is set to 0),
- the saturated model (i.e., the model with a deviance of 0), and
- the random/mixed-effects model (i.e., where τ^2 is estimated) (only if `method="ML"`).

The saturated model is obtained by adding as many dummy variables to the model as needed so that the model deviance is equal to zero. Even when `method="ML"`, the fixed-effects and saturated models are fitted, as they are used to compute the test statistics for the Wald-type and likelihood ratio tests for (residual) heterogeneity (see below).

Omnibus Test of Moderators

For models including moderators, an omnibus test of all model coefficients is conducted that excludes the intercept (the first coefficient) if it is included in the model. If no intercept is included in the model, then the omnibus test includes all of the coefficients in the model including the first. Alternatively, one can manually specify the indices of the coefficients to test via the `btt` argument. For example, with `btt=c(3,4)`, only the third and fourth coefficient from the model would be included in the test (if an intercept is included in the model, then it corresponds to the first coefficient in the model). Instead of specifying the coefficient numbers, one can specify a string for `btt`. In that case, `grep` will be used to search for all coefficient names that match the string. The omnibus test is called the Q_M-test and follows, under the assumptions of the model, a chi-square distribution with m degrees of freedom (with m denoting the number of coefficients tested) under the null hypothesis (that the true value of all coefficients tested is equal to 0).

Categorical Moderators

Categorical moderator variables can be included in the model via the `mods` argument in the same way that appropriately (dummy) coded categorical independent variables can be included in linear models. One can either do the dummy coding manually or use a model formula together with the `factor` function to let R handle the coding automatically (note that string/character variables in a model formula are automatically converted to factors).

Tests and Confidence Intervals

By default, tests of individual coefficients in the model (and the corresponding confidence intervals) are based on a standard normal distribution, while the omnibus test is based on a chi-square distribution (see above). As an alternative, one can set `test="t"`, in which case tests of individual coefficients and confidence intervals are based on a t-distribution with $k - p$ degrees of freedom, while the omnibus test statistic then uses an F-distribution with m and $k - p$ degrees of freedom (with k denoting the total number of estimates included in the analysis and p the total number of model coefficients including the intercept if it is present). Note that `test="t"` is not the same as `test="knha"` in `rma.uni`, as no adjustment to the standard errors of the estimated coefficients is made.

Tests for (Residual) Heterogeneity

Two different tests for (residual) heterogeneity are automatically carried out by the function. The first is a Wald-type test, which tests the coefficients corresponding to the dummy variables added in the saturated model for significance. The second is a likelihood ratio test, which tests the same set of coefficients, but does so by computing -2 times the difference in the log-likelihood of the fixed-effects and the saturated model. These two tests are not identical for the types of models fitted by the `rma.glmm` function and may even lead to conflicting conclusions.

Observed Effect Sizes or Outcomes of the Individual Studies

The various models do not require the calculation of the observed effect sizes or outcomes of the individual studies (e.g., the observed odds ratios of the k studies) and directly make use of the table/event counts. Zero cells/events are not a problem (except in extreme cases, such as when one of the two outcomes never occurs or when there are no events in any of the studies). Therefore, it is unnecessary to add some constant to the cell/event counts when there are zero cells/events.

However, for plotting and various other functions, it is necessary to calculate the observed effect sizes or outcomes for the k studies. Here, zero cells/events can be problematic, so adding a constant value to the cell/event counts ensures that all k values can be calculated. The `add` and `to` arguments are used to specify what value should be added to the cell/event counts and under what circumstances when calculating the observed effect sizes or outcomes. The documentation of the `escalc` function explains how the `add` and `to` arguments work. Note that `drop00` is set to `TRUE` by default, since studies where `ai=ci=0` or `bi=di=0` or studies where `x1i=x2i=0` are uninformative about the size of the effect.

Value

An object of class `c("rma.glmm","rma")`. The object is a list containing the following components:

`beta`	estimated coefficients of the model.
`se`	standard errors of the coefficients.
`zval`	test statistics of the coefficients.
`pval`	corresponding p-values.
`ci.lb`	lower bound of the confidence intervals for the coefficients.
`ci.ub`	upper bound of the confidence intervals for the coefficients.
`vb`	variance-covariance matrix of the estimated coefficients.
`tau2`	estimated amount of (residual) heterogeneity. Always `0` when `method="FE"`.
`sigma2`	estimated amount of study level variability (only for `model="UM.RS"`).
`k`	number of studies included in the analysis.
`p`	number of coefficients in the model (including the intercept).
`m`	number of coefficients included in the omnibus test of moderators.
`QE.Wld`	Wald-type test statistic of the test for (residual) heterogeneity.
`QEp.Wld`	corresponding p-value.
`QE.LRT`	likelihood ratio test statistic of the test for (residual) heterogeneity.
`QEp.LRT`	corresponding p-value.
`QM`	test statistic of the omnibus test of moderators.
`QMp`	corresponding p-value.
`I2`	value of I^2.
`H2`	value of H^2.
`int.only`	logical that indicates whether the model is an intercept-only model.
`yi, vi, X`	the vector of outcomes, the corresponding sampling variances, and the model matrix.
`fit.stats`	a list with the log-likelihood, deviance, AIC, BIC, and AICc values.
`...`	some additional elements/values.

Methods

The results of the fitted model are formatted and printed with the `print.rma.glmm` function. If fit statistics should also be given, use `summary.rma` (or use the `fitstats.rma` function to extract them).

Note

Fitting the various types of models requires several different iterative algorithms:

- For `model="UM.FS"` and `model="CM.AL"`, iteratively reweighted least squares (IWLS) as implemented in the `glm` function is used for fitting the fixed-effects and the saturated models. For `method="ML"`, adaptive Gauss-Hermite quadrature as implemented in the `glmer` function is used. The same applies when `model="CM.EL"` is used in combination with `measure="IRR"` or when `measure="PLO"` or `measure="IRLN"` (regardless of the model type).
- For `model="UM.RS"`, adaptive Gauss-Hermite quadrature as implemented in the `glmer` function is used to fit all of the models.
- For `model="CM.EL"` and `measure="OR"`, the quasi-Newton method (`"BFGS"`) as implemented in the `optim` function is used by default for fitting the fixed-effects and the saturated models. For `method="ML"`, the same algorithm is used, together with adaptive quadrature as implemented in the `integrate` function (for the integration over the density of the non-central hypergeometric distribution). Standard errors of the parameter estimates are obtained by inverting the Hessian, which is numerically approximated using the `hessian` function.

When `model="CM.EL"` and `measure="OR"`, actually `model="CM.AL"` is used first to obtain starting values for `optim`, so either 4 (if `method="FE"`) or 6 (if `method="ML"`) models need to be fitted in total.

Various control parameters can be adjusted via the `control` argument:

- `optimizer` is set by default to `"optim"`, but can be set to `"nlminb"` or one of the optimizers from the **minqa** package (i.e., `"bobyqa"`, `"newuoa"`, or `"uobyqa"`),
- `optmethod` is used to set the `method` argument for `optim` (the default is `"BFGS"`),
- `optCtrl` is a list of named arguments to be passed on to the `control` argument of the chosen optimizer,
- `glmCtrl` is a list of named arguments to be passed on to the `control` argument of the `glm` function,
- `glmerCtrl` is a list of named arguments to be passed on to the `control` argument of the `glmer` function, and
- `intCtrl` is a list of named arguments (i.e., `rel.tol` and `subdivisions`) to be passed on to the `integrate` function.
- `hessianCtrl` is a list of named arguments to be passed on to the `method.args` argument of the `hessian` function. For some borderline cases, it may be necessary to bump up the `r` argument to a higher number to get sufficient accuracy when approximating the Hessian numerically (the default is `control=list(hessianCtrl=list(r=16))`).

Also, for `glmer`, the `nAGQ` argument is used to specify the number of quadrature points. The default value is 7, which should provide sufficient accuracy in the evaluation of the log-likelihood in most cases, but at the expense of speed. Setting this to 1 corresponds to the Laplacian approximation (which is faster, but less accurate).

Information on the progress of the various algorithms can be obtained by setting `verbose=TRUE`. Since fitting the various models can be computationally expensive, this option is useful to determine how the model fitting is progressing. One can also set `verbose` to an integer (`verbose=2` yields even more information and `verbose=3` also sets `option(warn=1)` temporarily).

For `model="CM.EL"` and `measure="OR"`, optimization involves repeated calculation of the density of the non-central hypergeometric distribution. When `method="ML"`, this also requires integration over the same density. This is currently implemented in a rather brute-force manner and may not be numerically stable, especially when models with moderators are fitted. Stability can be improved by scaling the moderators in a similar manner (i.e., don't use a moderator that is coded 0 and 1, while another uses values in the 1000s). For models with an intercept and moderators, the function actually rescales (non-dummy) variables to z-scores during the model fitting (results are given after back-scaling, so this should be transparent to the user). For models without an intercept, this is not done, so sensitivity analyses are highly recommended here (to ensure that the results do not depend on the scaling of the moderators).

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

Code for computing the density of the non-central hypergeometric distribution comes from the MCMCpack package, which in turn is based on Liao and Rosen (2001).

References

Agresti, A. (2002). *Categorical data analysis* (2nd. ed). Hoboken, NJ: Wiley.

Bagos, P. G., & Nikolopoulos, G. K. (2009). Mixed-effects Poisson regression models for meta-analysis of follow-up studies with constant or varying durations. *The International Journal of Biostatistics*, **5**(1). https://doi.org/10.2202/1557-4679.1168

van Houwelingen, H. C., Zwinderman, K. H., & Stijnen, T. (1993). A bivariate approach to meta-analysis. *Statistics in Medicine*, **12**(24), 2273–2284. https://doi.org/10.1002/sim.4780122405

Liao, J. G., & Rosen, O. (2001). Fast and stable algorithms for computing and sampling from the noncentral hypergeometric distribution. *American Statistician*, **55**(4), 366–369. https://doi.org/10.1198/000313001753272547

Stijnen, T., Hamza, T. H., & Ozdemir, P. (2010). Random effects meta-analysis of event outcome in the framework of the generalized linear mixed model with applications in sparse data. *Statistics in Medicine*, **29**(29), 3046–3067. https://doi.org/10.1002/sim.4040

Turner, R. M., Omar, R. Z., Yang, M., Goldstein, H., & Thompson, S. G. (2000). A multilevel model framework for meta-analysis of clinical trials with binary outcomes. *Statistics in Medicine*, **19**(24), 3417–3432. https://doi.org/10.1002/1097-0258(20001230)19:24<3417::aid-sim614>3.0.co;2-l

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`rma.uni`, `rma.mh`, `rma.peto`, and `rma.mv` for other model fitting functions.

`dat.nielweise2007`, `dat.nielweise2008`, `dat.collins1985a`, and `dat.pritz1997` for further examples of the use of the `rma.glmm` function.

Examples

```
### random-effects model using rma.uni() (standard RE model analysis)
rma(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, method="ML")

### random-effects models using rma.glmm() (require 'lme4' package)

### unconditional model with fixed study effects
## Not run:
rma.glmm(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, model="UM.FS")
## End(Not run)

### unconditional model with random study effects
## Not run:
rma.glmm(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, model="UM.RS")
## End(Not run)

### conditional model with approximate likelihood
## Not run:
rma.glmm(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, model="CM.AL")
## End(Not run)

### conditional model with exact likelihood
### note: fitting this model may take a bit of time, so be patient
## Not run:
rma.glmm(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, model="CM.EL")
## End(Not run)
```

---

rma.mh *Meta-Analysis via the Mantel-Haenszel Method*

Description

Function to fit fixed-effects models to 2×2 table and person-time data via the Mantel-Haenszel method. See below and the documentation of the **metafor-package** for more details on these models.

Usage

```
rma.mh(ai, bi, ci, di, n1i, n2i, x1i, x2i, t1i, t2i,
       measure="OR", data, slab, subset,
       add=1/2, to="only0", drop00=TRUE,
       correct=TRUE, level=95, digits, verbose=FALSE, ...)
```

Arguments

ai	vector to specify the 2×2 table frequencies (upper left cell). See below and the documentation of the escalc function for more details.
bi	vector to specify the 2×2 table frequencies (upper right cell). See below and the documentation of the escalc function for more details.

ci vector to specify the 2×2 table frequencies (lower left cell). See below and the documentation of the escalc function for more details.

di vector to specify the 2×2 table frequencies (lower right cell). See below and the documentation of the escalc function for more details.

n1i vector to specify the group sizes or row totals (first group). See below and the documentation of the escalc function for more details.

n2i vector to specify the group sizes or row totals (second group). See below and the documentation of the escalc function for more details.

x1i vector to specify the number of events (first group). See below and the documentation of the escalc function for more details.

x2i vector to specify the number of events (second group). See below and the documentation of the escalc function for more details.

t1i vector to specify the total person-times (first group). See below and the documentation of the escalc function for more details.

t2i vector to specify the total person-times (second group). See below and the documentation of the escalc function for more details.

measure character string to specify the outcome measure to use for the meta-analysis. Possible options are the risk ratio ("RR"), the odds ratio ("OR"), the risk difference ("RD"), the incidence rate ratio ("IRR"), or the incidence rate difference ("IRD").

data optional data frame containing the data supplied to the function.

slab optional vector with labels for the k studies.

subset optional (logical or numeric) vector to specify the subset of studies that should be used for the analysis.

add non-negative number to specify the amount to add to zero cells, counts, or frequencies when calculating the observed effect sizes or outcomes of the individual studies. Can also be a vector of two numbers, where the first number is used in the calculation of the observed effect sizes or outcomes and the second number is used when applying the Mantel-Haenszel method. See below and the documentation of the escalc function for more details.

to character string to specify when the values under add should be added (either "only0", "all", "if0all", or "none"). Can also be a character vector, where the first string again applies when calculating the observed effect sizes or outcomes and the second string when applying the Mantel-Haenszel method. See below and the documentation of the escalc function for more details.

drop00 logical to specify whether studies with no cases/events (or only cases) in both groups should be dropped when calculating the observed effect sizes or outcomes (the outcomes for such studies are set to NA). Can also be a vector of two logicals, where the first applies to the calculation of the observed effect sizes or outcomes and the second when applying the Mantel-Haenszel method. See below and the documentation of the escalc function for more details.

correct logical to specify whether to apply a continuity correction when computing the Cochran-Mantel-Haenszel test statistic.

`level` numeric value between 0 and 100 to specify the confidence interval level (the default is 95).

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is 4.

`verbose` logical to specify whether output should be generated on the progress of the model fitting (the default is `FALSE`).

`...` additional arguments.

Details

Specifying the Data

When the outcome measure is either the risk ratio (measure="RR"), odds ratio (`measure="OR"`), or risk difference (`measure="RD"`), the studies are assumed to provide data in terms of 2 × 2 tables of the form:

	outcome 1	outcome 2	total
group 1	`ai`	`bi`	`n1i`
group 2	`ci`	`di`	`n2i`

where `ai`, `bi`, `ci`, and `di` denote the cell frequencies and `n1i` and `n2i` the row totals. For example, in a set of randomized clinical trials (RCTs) or cohort studies, group 1 and group 2 may refer to the treatment (exposed) and placebo/control (not exposed) group, with outcome 1 denoting some event of interest (e.g., death) and outcome 2 its complement. In a set of case-control studies, group 1 and group 2 may refer to the group of cases and the group of controls, with outcome 1 denoting, for example, exposure to some risk factor and outcome 2 non-exposure. For these outcome measures, one needs to specify either `ai`, `bi`, `ci`, and `di` or alternatively `ai`, `ci`, `n1i`, and `n2i`.

Alternatively, when the outcome measure is the incidence rate ratio (`measure="IRR"`) or the incidence rate difference (`measure="IRD"`), the studies are assumed to provide data in terms of tables of the form:

	events	person-time
group 1	`x1i`	`t1i`
group 2	`x2i`	`t2i`

where `x1i` and `x2i` denote the number of events in the first and the second group, respectively, and `t1i` and `t2i` the corresponding total person-times at risk.

Mantel-Haenszel Method

An approach for aggregating data of these types was suggested by Mantel and Haenszel (1959) and later extended by various authors (see references). The Mantel-Haenszel method provides a weighted estimate under a fixed-effects model. The method is particularly advantageous when aggregating a large number of studies with small sample sizes (the so-called sparse data or increasing strata case).

When analyzing odds ratios, the Cochran-Mantel-Haenszel (CMH) test (Cochran, 1954; Mantel & Haenszel, 1959) and Tarone's test for heterogeneity (Tarone, 1985) are also provided (by default, the CMH test statistic is computed with the continuity correction; this can be switched off with `correct=FALSE`). When analyzing incidence rate ratios, the Mantel-Haenszel (MH) test (Rothman et al., 2008) for person-time data is also provided (again, the `correct` argument controls whether the continuity correction is applied). When analyzing risk ratios, odds ratios, or incidence rate ratios, the printed results are given both in terms of the log and the raw units (for easier interpretation).

Observed Effect Sizes or Outcomes of the Individual Studies

The Mantel-Haenszel method itself does not require the calculation of the observed effect sizes or outcomes of the individual studies (e.g., the observed odds or incidence rate ratios of the k studies) and directly makes use of the table/event counts. Zero cells/events are not a problem (except in extreme cases, such as when one of the two outcomes never occurs in any of the 2×2 tables or when there are no events for one of the two groups in any of the tables). Therefore, it is unnecessary to add some constant to the cell/event counts when there are zero cells/events.

However, for plotting and various other functions, it is necessary to calculate the observed effect sizes or outcomes for the k studies. Here, zero cells/events can be problematic, so adding a constant value to the cell/event counts ensures that all k values can be calculated. The `add` and `to` arguments are used to specify what value should be added to the cell/event counts and under what circumstances when calculating the observed effect sizes or outcomes and when applying the Mantel-Haenszel method. Similarly, the `drop00` argument is used to specify how studies with no cases/events (or only cases) in both groups should be handled. The documentation of the `escalc` function explains how the `add`, `to`, and `drop00` arguments work. If only a single value for these arguments is specified (as per default), then these values are used when calculating the observed effect sizes or outcomes and no adjustment to the cell/event counts is made when applying the Mantel-Haenszel method. Alternatively, when specifying two values for these arguments, the first value applies when calculating the observed effect sizes or outcomes and the second value when applying the Mantel-Haenszel method.

Note that `drop00` is set to `TRUE` by default. Therefore, the observed effect sizes or outcomes for studies where `ai=ci=0` or `bi=di=0` or studies where `x1i=x2i=0` are set to `NA`. When applying the Mantel-Haenszel method, such studies are not explicitly dropped (unless the second value of `drop00` argument is also set to `TRUE`), but this is practically not necessary, as they do not actually influence the results (assuming no adjustment to the cell/event counts are made when applying the Mantel-Haenszel method).

Value

An object of class `c("rma.mh","rma")`. The object is a list containing the following components:

`beta`	aggregated log risk ratio, log odds ratio, risk difference, log rate ratio, or rate difference.
`se`	standard error of the aggregated value.
`zval`	test statistics of the aggregated value.
`pval`	corresponding p-value.
`ci.lb`	lower bound of the confidence interval.
`ci.ub`	upper bound of the confidence interval.
`QE`	test statistic of the test for heterogeneity.
`QEp`	correspinding p-value.
`MH`	Cochran-Mantel-Haenszel test statistic (`measure="OR"`) or Mantel-Haenszel test statistic (`measure="IRR"`).

MHp	corresponding p-value.
TA	test statistic of Tarone's test for heterogeneity (only when measure="OR").
TAp	corresponding p-value (only when measure="OR").
k	number of studies included in the analysis.
yi, vi	the vector of outcomes and corresponding sampling variances.
fit.stats	a list with the log-likelihood, deviance, AIC, BIC, and AICc values under the unrestricted and restricted likelihood.
...	some additional elements/values.

Methods

The results of the fitted model are formatted and printed with the `print.rma.mh` function. If fit statistics should also be given, use `summary.rma` (or use the `fitstats.rma` function to extract them).

The `residuals.rma`, `rstandard.rma.mh`, and `rstudent.rma.mh` functions extract raw and standardized residuals. Leave-one-out diagnostics can be obtained with `leave1out.rma.mh`.

Forest, funnel, radial, L'Abbé, and Baujat plots can be obtained with `forest.rma`, `funnel.rma`, `radial.rma`, `labbe.rma`, and `baujat.rma`. The `qqnorm.rma.mh` function provides normal QQ plots of the standardized residuals. One can also just call `plot.rma.mh` on the fitted model object to obtain various plots at once.

A cumulative meta-analysis (i.e., adding one observation at a time) can be obtained with `cumul.rma.mh`.

Other extractor functions include `coef.rma`, `vcov.rma`, `logLik.rma`, `deviance.rma`, `AIC.rma`, and `BIC.rma`.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Cochran, W. G. (1954). Some methods for strengthening the common χ^2 tests. *Biometrics*, **10**(4), 417–451. https://doi.org/10.2307/3001616

Greenland, S., & Robins, J. M. (1985). Estimation of a common effect parameter from sparse follow-up data. *Biometrics*, **41**(1), 55–68. https://doi.org/10.2307/2530643

Mantel, N., & Haenszel, W. (1959). Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the National Cancer Institute*, **22**(4), 719–748. https://doi.org/10.1093/jnci/22.4.719

Nurminen, M. (1981). Asymptotic efficiency of general noniterative estimators of common relative risk. *Biometrika*, **68**(2), 525–530. https://doi.org/10.1093/biomet/68.2.525

Robins, J., Breslow, N., & Greenland, S. (1986). Estimators of the Mantel-Haenszel variance consistent in both sparse data and large-strata limiting models. *Biometrics*, **42**(2), 311–323. https://doi.org/10.2307/2531052

Rothman, K. J., Greenland, S., & Lash, T. L. (2008). *Modern epidemiology* (3rd ed.). Philadelphia: Lippincott Williams & Wilkins.

Sato, T., Greenland, S., & Robins, J. M. (1989). On the variance estimator for the Mantel-Haenszel risk difference. *Biometrics*, **45**(4), 1323–1324. https://www.jstor.org/stable/2531784

Tarone, R. E. (1981). On summary estimators of relative risk. *Journal of Chronic Diseases*, **34**(9-10), 463–468. https://doi.org/10.1016/0021-9681(81)90006-0

Tarone, R. E. (1985). On heterogeneity tests based on efficient scores. *Biometrika*, **72**(1), 91–95. https://doi.org/10.1093/biomet/72.1.91

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`rma.uni`, `rma.glmm`, `rma.peto`, and `rma.mv` for other model fitting functions.

Examples

```
### meta-analysis of the (log) odds ratios using the Mantel-Haenszel method
rma.mh(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### meta-analysis of the (log) risk ratios using the Mantel-Haenszel method
rma.mh(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)
```

`rma.mv` *Meta-Analysis via Multivariate/Multilevel Linear (Mixed-Effects) Models*

Description

Function to fit meta-analytic multivariate/multilevel fixed- and random/mixed-effects models with or without moderators via linear (mixed-effects) models. See below and the documentation of the **metafor-package** for more details on these models.

Usage

```
rma.mv(yi, V, W, mods, random, struct="CS", intercept=TRUE,
        data, slab, subset, method="REML", test="z", dfs="residual",
        level=95, digits, btt, R, Rscale="cor",
       sigma2, tau2, rho, gamma2, phi, sparse=FALSE, verbose=FALSE, control, ...)
```

Arguments

`yi` vector of length k with the observed effect sizes or outcomes. See ‘Details’.

`V` vector of length k with the corresponding sampling variances or a $k \times k$ variance-covariance matrix of the sampling errors. See ‘Details’.

`W` optional argument to specify a vector of length k with user-defined weights or a $k \times k$ user-defined weight matrix. See ‘Details’.

`mods` optional argument to include one or more moderators in the model. A single moderator can be given as a vector of length k specifying the values of the moderator. Multiple moderators are specified by giving a matrix with k rows and as many columns as there are moderator variables. Alternatively, a model `formula` can be used to specify the model. See ‘Details’.

random — either a single one-sided formula or list of one-sided formulas to specify the random-effects structure of the model. See 'Details'.

struct — character string to specify the variance structure of an ~ inner | outer formula in the random argument. Either "CS" for compound symmetry, "HCS" for heteroscedastic compound symmetry, "UN" for an unstructured variance-covariance matrix, "ID" for a scaled identity matrix, "DIAG" for a diagonal matrix, "AR" for an AR(1) autoregressive structure, "HAR" for a heteroscedastic AR(1) autoregressive structure, "CAR" for a continuous-time autoregressive structure, or one of "SPEXP", "SPGAU", "SPLIN", "SPRAT", or "SPSPH" for one of the spatial correlation structures. See 'Details'.

intercept — logical to specify whether an intercept should be added to the model (the default is TRUE). Ignored when mods is a formula.

data — optional data frame containing the data supplied to the function.

slab — optional vector with labels for the k outcomes/studies.

subset — optional (logical or numeric) vector to specify the subset of studies (or more precisely, rows of the dataset) that should be used for the analysis.

method — character string to specify whether the model should be fitted via maximum-likelihood ("ML") or via restricted maximum-likelihood ("REML") estimation. Default is "REML".

test — character string to specify how test statistics and confidence intervals for the fixed effects should be computed. By default (test="z"), Wald-type tests and CIs are obtained, which are based on a standard normal distribution. When test="t", a t-distribution is used instead. See 'Details'.

dfs — character string to specify how the (denominator) degrees of freedom should be calculated when test="t". Either dfs="residual" or dfs="contain". Can also be a numeric vector with the degrees of freedom for each model coefficient. See 'Details'.

level — numeric value between 0 and 100 to specify the confidence interval level (the default is 95).

digits — integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is 4.

btt — optional vector of indices to specify which coefficients to include in the omnibus test of moderators. Can also be a string to grep for. See 'Details'.

R — an optional named list of known correlation matrices corresponding to (some of) the components specified via the random argument. See 'Details'.

Rscale — character string, integer, or logical to specify how matrices specified via the R argument should be scaled. See 'Details'.

sigma2 — optional numeric vector (of the same length as the number of random intercept components specified via the random argument) to fix the corresponding σ^2 value(s). A specific σ^2 value can be fixed by setting the corresponding element of this argument to the desired value. A specific σ^2 value will be estimated if the corresponding element is set equal to NA. See 'Details'.

tau2 — optional numeric value (for struct="CS", "AR", "CAR", or a spatial correlation structure) or vector (for struct="HCS", "UN", or "HAR") to fix the amount of (residual) heterogeneity for the levels of the inner factor corresponding to an `~ inner | outer` formula specified in the `random` argument. A numeric value fixes a particular τ^2 value, while `NA` means that the value should be estimated. See ‘Details’.

`rho` optional numeric value (for `struct="CS"`, `"HCS"`, `"AR"`, `"HAR"`, `"CAR"`, or a spatial correlation structure) or vector (for `struct="UN"`) to fix the correlation between the levels of the inner factor corresponding to an `~ inner | outer` formula specified in the `random` argument. A numeric value fixes a particular ρ value, while `NA` means that the value should be estimated. See ‘Details’.

`gamma2` as `tau2` argument, but for a second `~ inner | outer` formula specified in the `random` argument. See ‘Details’.

`phi` as `rho` argument, but for a second `~ inner | outer` formula specified in the `random` argument. See ‘Details’.

`sparse` logical to specify whether the function should use sparse matrix objects to the extent possible (can speed up model fitting substantially for certain models). See ‘Note’.

`verbose` logical to specify whether output should be generated on the progress of the model fitting (the default is `FALSE`). Can also be an integer. Values > 1 generate more verbose output. See ‘Note’.

`control` optional list of control values for the estimation algorithms. If unspecified, default values are defined inside the function. See ‘Note’.

`...` additional arguments.

Details

Specifying the Data

The function can be used in conjunction with any of the usual effect size or outcome measures used in meta-analyses (e.g., log risk ratios, log odds ratios, risk differences, mean differences, standardized mean differences, log transformed ratios of means, raw correlation coefficients, correlation coefficients transformed with Fisher’s r-to-z transformation, and so on). Simply specify the observed effect sizes or outcomes via the `yi` argument and the corresponding sampling variances via the `V` argument. In case the sampling errors are correlated, then one can specify the entire variance-covariance matrix of the sampling errors via the `V` argument.

The `escalc` function can be used to compute a wide variety of effect size or outcome measures (and the corresponding sampling variances) based on summary statistics. Equations for computing the covariance between the sampling errors for a variety of different effect size or outcome measures can be found, for example, in Gleser and Olkin (2009). For raw and Fisher r-to-z transformed correlations, one can find suitable equations, for example, in Steiger (1980). The latter are implemented in the `rcalc` function.

Specifying Fixed Effects

With `rma.mv(yi,V)`, a fixed-effects model is fitted to the data (note: arguments `struct`, `sigma2`, `tau2`, `rho`, `gamma2`, `phi`, `R`, and `Rscale` are not relevant then and are ignored). The model is then simply given by $y \sim N(\theta, V)$, where y is a (column) vector with the observed outcomes, θ is the (average) true outcome, and V is the variance-covariance matrix of the sampling errors (if a vector of sampling variances is provided via the `V` argument, then V is assumed to be diagonal).

One or more moderators can be included in the model via the mods argument. A single moderator can be given as a (row or column) vector of length k specifying the values of the moderator. Multiple moderators are specified by giving an appropriate model matrix (i.e., X) with k rows and as many columns as there are moderator variables (e.g., mods = cbind(mod1,mod2,mod3), where mod1, mod2, and mod3 correspond to the names of the variables for the three moderator variables). The intercept is added to the model matrix by default unless intercept=FALSE.

Alternatively, one can use standard formula syntax to specify the model. In this case, the mods argument should be set equal to a one-sided formula of the form mods = ~ model (e.g., mods = ~ mod1 + mod2 + mod3). Interactions, polynomial terms, and factors can be easily added to the model in this manner. When specifying a model formula via the mods argument, the intercept argument is ignored. Instead, the inclusion/exclusion of the intercept is controlled by the specified formula (e.g., mods = ~ mod1 + mod2 + mod3 -1 would lead to the removal of the intercept). One can also directly specify moderators via the yi argument (e.g., rma.mv(yi ~ mod1 + mod2 + mod3,V)). In that case, the mods argument is ignored and the inclusion/exclusion of the intercept again is controlled by the specified formula.

With moderators included, the model is then given by $y \sim N(X\beta, V)$, where X denotes the model matrix containing the moderator values (and possibly the intercept) and β is a column vector containing the corresponding model coefficients. The model coefficients (i.e., β) are then estimated with $b = (X'WX')^{-1}X'Wy$, where $W = V^{-1}$ is the weight matrix (without moderators, X is just a column vector of 1's). With the W argument, one can also specify user-defined weights (or a weight matrix).

Specifying Random Effects

One can fit random/mixed-effects models to the data by specifying the desired random effects structure via the random argument. The random argument is either a single one-sided formula or a list of one-sided formulas. One formula type that can be specified via this argument is of the form random = ~ 1 | id. Such a formula adds a random effect corresponding to the grouping variable/factor id to the model. Outcomes with the same value/level of the id variable/factor receive the same value of the random effect, while outcomes with different values/levels of the id variable/factor are assumed to be independent. The variance component corresponding to such a formula is denoted by σ^2. An arbitrary number of such formulas can be specified as a list of formulas (e.g., random = list(~ 1 | id1,~ 1 | id2)), with variance components σ^2_1, σ^2_2, and so on. Nested random effects of this form can also be added using random = ~ 1 | id1/id2, which adds a random effect corresponding to the grouping variable/factor id1 and a random effect corresponding to id2 within id1 to the model. This can be extended to models with even more levels of nesting (e.g., random = ~ 1 | id1/id2/id2).

Random effects of this form are useful to model clustering (and hence non-independence) induced by a multilevel structure in the data (e.g., outcomes derived from the same paper, lab, research group, or species may be more similar to each other than outcomes derived from different papers, labs, research groups, or species). See, for example, Konstantopoulos (2011) and Nakagawa and Santos (2012) for more details.

See dat.konstantopoulos2011, dat.bornmann2007, dat.obrien2003, and dat.crede2010 for examples of multilevel meta-analyses.

In addition or alternatively to specifying one or multiple ~ 1 | id terms, the random argument can also contain a formula of the form ~ inner | outer. Outcomes with the same value/level of the outer grouping variable/factor share correlated random effects corresponding to the levels of the inner grouping variable/factor, while outcomes with different values/levels of the outer grouping variable/factor are assumed to be independent (note that the inner grouping variable must either be a factor or a character variable). The `struct` argument is used to specify the variance structure corresponding to the `inner` variable/factor. With `struct="CS"`, a compound symmetric structure is assumed (i.e., a single variance component τ^2 corresponding to all values/levels of the `inner` variable/factor and a single correlation coefficient ρ for the correlation between the different values/levels). With `struct="HCS"`, a heteroscedastic compound symmetric structure is assumed (with variance components τ_1^2, τ_2^2, and so on, corresponding to the values/levels of the `inner` variable/factor and a single correlation coefficient ρ for the correlation between the different values/levels). With `struct="UN"`, an unstructured variance-covariance matrix is assumed (with variance components τ_1^2, τ_2^2, and so on, corresponding to the values/levels of the `inner` variable/factor and correlation coefficients ρ_{12}, ρ_{13}, ρ_{23}, and so on, for the various combinations of the values/levels of the `inner` variable/factor). For example, for an `inner` grouping variable/factor with four levels, the three structures correspond to variance-covariance matrices of the form:

`struct="CS"`

$$\begin{bmatrix} \tau^2 & \rho\tau^2 & \rho\tau^2 & \rho\tau^2 \\ & \tau^2 & \rho\tau^2 & \rho\tau^2 \\ & & \tau^2 & \rho\tau^2 \\ & & & \tau^2 \end{bmatrix}$$

`struct="HCS"`

$$\begin{bmatrix} \tau_1^2 & \rho\tau_1\tau_2 & \rho\tau_1\tau_3 & \rho\tau_1\tau_4 \\ & \tau_2^2 & \rho\tau_2\tau_3 & \rho\tau_2\tau_4 \\ & & \tau_3^2 & \rho\tau_3\tau_4 \\ & & & \tau_4^2 \end{bmatrix}$$

`struct="UN"`

$$\begin{bmatrix} \tau_1^2 & \rho_{12}\tau_1\tau_2 & \rho_{13}\tau_1\tau_3 & \rho_{14}\tau_1\tau_4 \\ & \tau_2^2 & \rho_{23}\tau_2\tau_3 & \rho_{24}\tau_2\tau_4 \\ & & \tau_3^2 & \rho_{34}\tau_3\tau_4 \\ & & & \tau_4^2 \end{bmatrix}$$

Structures `struct="ID"` and `struct="DIAG"` are just like `struct="CS"` and `struct="HCS"`, respectively, except that ρ is automatically set to 0, so that we either get a scaled identity matrix or a diagonal matrix.

With the `outer` factor corresponding to a study identification variable and the `inner` factor corresponding to a variable indicating the treatment type or study arm, such a random effect could be used to estimate how strongly different treatment effects or outcomes within the same study are correlated and/or whether the amount of heterogeneity differs across different treatment types/arms. Network meta-analyses (also known as mixed treatment comparisons) will also typically require such a random effect (e.g., Salanti et al., 2008). The meta-analytic bivariate model (e.g., van Houwelingen, Arends, & Stijnen, 2002) can also be fitted in this manner (see the examples below). The `inner` factor could also correspond to a variable indicating different types of outcomes measured within the same study, which allows for fitting multivariate models with multiple correlated effects/outcomes per study (e.g., Berkey et al., 1998; Kalaian & Raudenbush, 1996).

See `dat.berkey1998`, `dat.assink2016`, `dat.kalaian1996`, `dat.dagostino1998`, and `dat.craft2003` for examples of multivariate meta-analyses with multiple outcomes. See `dat.kearon1998` for an example using a bivariate model to analyze sensitivity and specificity. See `dat.hasselblad1998`, `dat.pagliaro1992`, `dat.lopez2019`, and `dat.senn2013` for examples of network meta-analyses.

For meta-analyses of studies reporting outcomes at multiple time points, it may also be reasonable to assume that the true effects/outcomes are correlated over time according to an autoregressive structure (Ishak et al., 2007; Trikalinos & Olkin, 2012). For this purpose, one can also choose `struct="AR"`, corresponding to a structure with a single variance component τ^2 and AR(1) autocorrelation among the values of the random effect. The values of the `inner` variable (which does not have to be a factor here) should then reflect the various time points, with increasing values reflecting later time points. This structure assumes equally spaced time points, so the actual values of the `inner` variable are not relevant, only their ordering. One can also use `struct="HAR"`, which allows for fitting a heteroscedastic AR(1) structure (with variance components τ_1^2, τ_2^2, and so on). Finally, when time points are not evenly spaced, one might consider using `struct="CAR"` for a continuous-time autoregressive structure. For example, for an `inner` grouping variable with four time points, these structures correspond to variance-covariance matrices of the form:

`struct="AR"` `struct="HAR"` `struct="CAR"`

$$\begin{bmatrix} \tau^2 & \rho\tau^2 & \rho^2\tau^2 & \rho^3\tau^2 \\ & \tau^2 & \rho\tau^2 & \rho^2\tau^2 \\ & & \tau^2 & \rho\tau^2 \\ & & & \tau^2 \end{bmatrix} \begin{bmatrix} \tau_1^2 & \rho\tau_1\tau_2 & \rho^2\tau_1\tau_3 & \rho^3\tau_1\tau_4 \\ & \tau_2^2 & \rho\tau_2\tau_3 & \rho^2\tau_2\tau_4 \\ & & \tau_3^2 & \rho\tau_3\tau_4 \\ & & & \tau_4^2 \end{bmatrix} \begin{bmatrix} \tau^2 & \tau^2\rho^{|t_1-t_2|} & \tau^2\rho^{|t_1-t_3|} & \tau^2\rho^{|t_1-t_4|} \\ & \tau^2 & \tau^2\rho^{|t_2-t_3|} & \tau^2\rho^{|t_2-t_4|} \\ & & \tau^2 & \tau^2\rho^{|t_3-t_4|} \\ & & & \tau^2 \end{bmatrix}$$

See `dat.fine1993` and `dat.ishak2007` for examples involving such structures.

For outcomes that have a known spatial configuration, various spatial correlation structures are also available. For these structures, the formula is of the form `random = ~ var1 + var2 + ... | outer`, where `var1`, `var2`, and so on are variables to indicate the spatial coordinates (e.g., longitude and latitude) based on which distances (by default Euclidean) will be computed. Let d denote the distance between two points that share the same level of the `outer` variable (if all true effects/outcomes are allowed to be spatially correlated, simply set `outer` to a variable that is a constant). Then the correlation between the true effects/outcomes corresponding to these two points is a function of d and the parameter ρ. The following table shows the types of spatial correlation structures that can be specified and the equations for the correlation. The covariance between the true effects/outcomes is then the correlation times τ^2.

structure	struct	correlation
exponential	`"SPEXP"`	$\exp(-d/\rho)$
Gaussian	`"SPGAU"`	$\exp(-d^2/\rho^2)$
linear	`"SPLIN"`	$(1-d/\rho)I(d<\rho)$
rational quadratic	`"SPRAT"`	$1-(d/\rho)^2/(1+(d/\rho)^2)$
spherical	`"SPSPH"`	$(1-1.5(d/\rho)+0.5(d/\rho)^3)I(d<\rho)$

Note that $I(d < \rho)$ is equal to 1 if $d < \rho$ and 0 otherwise. The parameterization of the various structures is based on Pinheiro and Bates (2000). Instead of Euclidean distances, one can also use other distance measures by setting (the undocumented) argument `dist` to either `"maximum"` for the maximum distance between two points (supremum norm), to `"manhattan"` for the absolute distance between the coordinate vectors (L1 norm), or to `"gcd"` for the great-circle distance (WGS84 ellipsoid method). In the latter case, only two variables, namely the longitude and latitude (in decimal degrees, with minus signs for West and South), must be specified.

If a distance matrix has already been computed, one can also pass this matrix as a list element to the `dist` argument. In this case, one should use a formula of the form `random = ~ id | outer`, where `id` are location identifiers, with corresponding row/column names in the distance matrix specified via the `dist` argument.

See `dat.maire2019` for an example of a meta-analysis with a spatial correlation structure.

The `random` argument can also contain a second formula of the form `~ inner | outer` (but no more!). A second formula of this form works exactly described as above, but its variance components are denoted by γ^2 and its correlation components by ϕ. The `struct` argument should then be of length 2 to specify the variance-covariance structure for the first and second component, respectively.

When the `random` argument contains a formula of the form `~ 1 | id`, one can use the (optional) argument `R` to specify a corresponding known correlation matrix for the random effect (i.e., `R = list(id = Cor)`, where `Cor` is the correlation matrix). In that case, outcomes with the same value/level of the `id` variable/factor receive the same value for the random effect, while outcomes with different values/levels of the `id` variable/factor receive values that are correlated as specified in the corresponding correlation matrix given via the `R` argument. The column/row names of the correlation matrix given via the `R` argument must therefore contain all of the values/levels of the `id` variable/factor. When the `random` argument contains multiple formulas of the form `~ 1 | id`, one can specify known correlation matrices for none, some, or all of those terms (e.g., with `random = list(~ 1 | id1, ~ 1 | id2)`, one could specify `R = list(id1 = Cor1)` or `R = list(id1 = Cor1, id2 = Cor2)`, where `Cor1` and `Cor2` are the correlation matrices corresponding to the grouping variables/factors `id1` and `id2`, respectively).

Such a random effect with a known (or at least approximately known) correlation structure is useful in a variety of contexts. For example, such a component can be used to account for the correlations induced by the shared phylogenetic history among organisms (e.g., plants, fungi, animals). In that case, `~ 1 | species` is used to specify the species and argument `R` is used to specify the phylogenetic correlation matrix of the species studied in the meta-analysis. The corresponding variance component then indicates how much variance/heterogeneity is attributable to the specified phylogeny. See Nakagawa and Santos (2012) for more details. As another example, in a genetic meta-analysis studying disease association for several single nucleotide polymorphisms (SNPs), linkage disequilibrium (LD) among the SNPs can induce an approximately known degree of correlation among the effects/outcomes. In that case, `~ 1 | snp` could be used to specify the SNPs and `R` the corresponding LD correlation matrix for the SNPs included in the meta-analysis.

The `Rscale` argument controls how matrices specified via the `R` argument are scaled. With `Rscale="none"` (or `Rscale=0` or `Rscale=FALSE`), no scaling is used. With `Rscale="cor"` (or `Rscale=1` or `Rscale=TRUE`), the `cov2cor` function is used to ensure that the matrices are correlation matrices (assuming they were covariance matrices to begin with). With `Rscale="cor0"` (or `Rscale=2`), first `cov2cor` is used and then the elements of each correlation matrix are scaled with $(R - \min(R))/(1 - \min(R))$ (this ensures that a correlation of zero in a phylogenetic correlation matrix corresponds to the split at the root node of the tree comprising the species that are actually analyzed). Finally, `Rscale="cov0"` (or `Rscale=3`) only rescales with $(R - \min(R))$ (which ensures that a phylogenetic covariance matrix is rooted at the lowest split).

See `dat.moura2021` and `dat.lim2014` for examples of meta-analyses with phylogenetic correlation structures.

Together with the variance-covariance matrix of the sampling errors (i.e., V), the specified random effects structure of the model implies a particular marginal variance-covariance matrix of the observed effect sizes or outcomes. Once estimates of the variance components (i.e., σ^2, τ^2, ρ, γ^2, and/or ϕ, values) have been obtained (either using maximum likelihood or restricted maximum likelihood estimation), the estimated marginal variance-covariance matrix can be constructed (denoted by M). The model coefficients (i.e., β) are then estimated with $b = (X'WX')^{-1}X'Wy$, where $W = M^{-1}$ is the weight matrix. With the `W` argument, one can again specify user-defined weights (or a weight matrix).

Fixing Variance Components and/or Correlations

Arguments `sigma2`, `tau2`, `rho`, `gamma2`, and `phi` can be used to fix particular variance components and/or correlations at a given value. This is useful for sensitivity analyses (e.g., for plotting the regular/restricted log-likelihood as a function of a particular variance component or correlation) or for imposing a desired variance-covariance structure on the data.

For example, if `random = list(~ 1 | id1, ~ 1 | id2)`, then `sigma2` must be of length 2 (corresponding to σ^2_1 and σ^2_2) and a fixed value can be assigned to either or both variance components. Setting a particular component to `NA` means that the component will be estimated by the function (e.g., `sigma2=c(0,NA)` would fix σ^2_1 to 0 and estimate σ^2_2).

Argument tau2 is only relevant when the random argument contains an ~ inner | outer formula. In that case, if the tau2 argument is used, it must be either of length 1 (for "CS", "ID", "AR", "CAR", or one of the spatial correlation structures) or of the same length as the number of levels of the inner factor (for "HCS", "DIAG", "UN", or "HAR"). A numeric value in the tau2 argument then fixes the corresponding variance component to that value, while NA means that the component will be estimated. Similarly, if argument rho is used, it must be either of length 1 (for "CS", "HCS", "AR", "HAR", or one of the spatial correlation structures) or of length $lvls(lvls-1)/2$ (for "UN"), where $lvls$ denotes the number of levels of the inner factor. Again, a numeric value fixes the corresponding correlation, while NA means that the correlation will be estimated. For example, with struct="CS" and rho=0, the variance-covariance matrix of the inner factor will be diagonal with τ^2 along the diagonal. For struct="UN", the values specified under rho should be given in column-wise order (e.g., for an inner grouping variable/factor with four levels, the order would be ρ_{12}, ρ_{13}, ρ_{23}, ρ_{14}, ρ_{24}, ρ_{34}.

Similarly, arguments gamma2 and phi are only relevant when the random argument contains a second ~ inner | outer formula. The arguments then work exactly as described above.

Omnibus Test of Moderators

For models including moderators, an omnibus test of all model coefficients is conducted that excludes the intercept (the first coefficient) if it is included in the model. If no intercept is included in the model, then the omnibus test includes all of the coefficients in the model including the first. Alternatively, one can manually specify the indices of the coefficients to test via the btt argument. For example, with btt=c(3,4), only the third and fourth coefficient from the model would be included in the test (if an intercept is included in the model, then it corresponds to the first coefficient in the model). Instead of specifying the coefficient numbers, one can specify a string for btt. In that case, grep will be used to search for all coefficient names that match the string. The omnibus test is called the Q_M-test and follows, under the assumptions of the model, a chi-square distribution with m degrees of freedom (with m denoting the number of coefficients tested) under the null hypothesis (that the true value of all coefficients tested is equal to 0).

Categorical Moderators

Categorical moderator variables can be included in the model via the mods argument in the same way that appropriately (dummy) coded categorical independent variables can be included in linear models. One can either do the dummy coding manually or use a model formula together with the factor function to let R handle the coding automatically (note that string/character variables in a model formula are automatically converted to factors).

Tests and Confidence Intervals

By default, tests of individual coefficients in the model (and the corresponding confidence intervals) are based on a standard normal distribution, while the omnibus test is based on a chi-square distribution (see above). As an alternative, one can set test="t", in which case tests of individual coefficients and confidence intervals are based on a t-distribution with $k-p$ degrees of freedom, while the omnibus test statistic then uses an F-distribution with m and $k-p$ degrees of freedom (with k denoting the total number of estimates included in the analysis and p the total number of model coefficients including the intercept if it is present). Note that test="t" is not the same as test="knha" in rma.uni, as no adjustment to the standard errors of the estimated coefficients is made.

The method for calculating the (denominator) degrees of freedom described above (which corresponds to dfs="residual") is quite simplistic and may lead to tests with inflated Type I error rates and confidence intervals that are too narrow on average. As an alternative, one can set `dfs="contain"`, in which case the degrees of freedom for the test of a particular model coefficient, b_j, are determined by checking whether x_j, the corresponding column of the model matrix X, varies at the level corresponding to a particular random effect in the model. If such a random effect can be found, then the degrees of freedom are set to $l - p$, where l denotes the unique number of values/levels that the random effect can take on (i.e., for an `~ 1 | id` term, the unique number of values/levels of the `id` variable and for an `~ inner | outer` term, the unique number of values/levels of the `outer` variable). If no such random effect can be found, then $k - p$ is used as the degrees of freedom. For the omnibus F-test, the minimum of the degrees of freedom of all coefficients involved in the test is used as the denominator degrees of freedom. This approach for calculating the degrees of freedom should often lead to tests with better control of the Type I error rate and confidence intervals with closer to nominal coverage rates.

One can also set `dfs` to a numeric vector with the desired values for the degrees of freedom for testing the model coefficients (e.g., if some other method for determining the degrees of freedom was used).

Test for (Residual) Heterogeneity

A test for (residual) heterogeneity is automatically carried out by the function. Without moderators in the model, this test is the generalized/weighted least squares extension of Cochran's Q-test, which tests whether the variability in the observed effect sizes or outcomes is larger than one would expect based on sampling variability (and the given covariances among the sampling errors) alone. A significant test suggests that the true effects/outcomes are heterogeneous. When moderators are included in the model, this is the Q_E-test for residual heterogeneity, which tests whether the variability in the observed effect sizes or outcomes that is not accounted for by the moderators included in the model is larger than one would expect based on sampling variability (and the given covariances among the sampling errors) alone.

Value

An object of class `c("rma.mv","rma")`. The object is a list containing the following components:

`beta`	estimated coefficients of the model.
`se`	standard errors of the coefficients.
`zval`	test statistics of the coefficients.
`pval`	corresponding p-values.
`ci.lb`	lower bound of the confidence intervals for the coefficients.
`ci.ub`	upper bound of the confidence intervals for the coefficients.
`vb`	variance-covariance matrix of the estimated coefficients.
`sigma2`	estimated σ^2 value(s).
`tau2`	estimated τ^2 value(s).
`rho`	estimated ρ value(s).
`gamma2`	estimated γ^2 value(s).
`phi`	estimated ϕ value(s).
`k`	number of observed effect sizes or outcomes included in the analysis.
`p`	number of coefficients in the model (including the intercept).
`m`	number of coefficients included in the omnibus test of moderators.

QE test statistic of the test for (residual) heterogeneity.

QEp corresponding p-value.

QM test statistic of the omnibus test of moderators.

QMp corresponding p-value.

int.only logical that indicates whether the model is an intercept-only model.

yi, V, X the vector of outcomes, the corresponding variance-covariance matrix of the sampling errors, and the model matrix.

M the estimated marginal variance-covariance matrix of the observed effect sizes or outcomes.

fit.stats a list with the log-likelihood, deviance, AIC, BIC, and AICc values.

... some additional elements/values.

Methods

The results of the fitted model are formatted and printed with the print.rma.mv function. If fit statistics should also be given, use summary.rma (or use the fitstats.rma function to extract them). Full versus reduced model comparisons in terms of fit statistics and likelihoods can be obtained with anova.rma. Wald-type tests for sets of model coefficients or linear combinations thereof can be obtained with the same function. Tests and confidence intervals based on (cluster) robust methods can be obtained with robust.rma.mv.

Predicted/fitted values can be obtained with predict.rma and fitted.rma. For best linear unbiased predictions, see ranef.rma.mv.

The residuals.rma, rstandard.rma.mv, and rstudent.rma.mv functions extract raw and standardized residuals. See influence.rma.mv for additional case diagnostics (e.g., to determine influential studies). For models with moderators, variance inflation factors can be obtained with vif.rma.

Confidence intervals for any variance/correlation parameters in the model can be obtained with confint.rma.mv.

For random/mixed-effects models, the profile.rma.mv function can be used to obtain a plot of the (restricted) log-likelihood as a function of a specific variance component or correlation parameter of the model. For models with moderators, regplot.rma draws scatter plots / bubble plots, showing the (marginal) relationship between the observed outcomes and a selected moderator from the model.

Other extractor functions include coef.rma, vcov.rma, logLik.rma, deviance.rma, AIC.rma, BIC.rma, hatvalues.rma.mv, and weights.rma.mv.

Note

Argument V also accepts a list of variance-covariance matrices for the observed effect sizes or outcomes. From the list elements, the full (block diagonal) variance-covariance matrix is then automatically constructed.

Model fitting is done via numerical optimization over the model parameters. By default, nlminb is used for the optimization. One can also chose a different optimizer via the control argument (e.g., control=list(optimizer="optim")). When using optim, one can set the particular method via the optmethod argument (e.g., control=list(optimizer="optim",optmethod="BFGS")). Besides nlminb and optim, one can also choose one of the optimizers from the minqa package (i.e., uobyqa, newuoa, or bobyqa), one of the (derivative-free) algorithms from the nloptr package, the Newton-type algorithm implemented in nlm, the various algorithms implemented in the dfoptim package (hjk for the Hooke-Jeeves, nmk for the Nelder-Mead, and mads for the Mesh Adaptive Direct Searches (MADS) algorithm), the quasi-Newton type optimizer ucminf from the package of the same name, or the parallelized version of the L-BFGS-B algorithm implemented in optimParallel from the package of the same name.

The optimizer name must be given as a character string (i.e., in quotes). Additional control parameters can be specified via the control argument (e.g., control=list(iter.max=500,rel.tol=1e-8)). For nloptr, the default is to use the BOBYQA implementation from that package with a relative convergence criterion of 1e-8 on the function value (i.e., log-likelihood), but this can be changed via the algorithm and ftop_rel arguments (e.g., control=list(optimizer="nloptr",algorithm="NLOPT_LN_SBPLX",fto For optimParallel, the control argument ncpus can be used to specify the number of cores to use for the parallelization (e.g., control=list(optimizer="optimParallel",ncpus=2)). With parallel::detectCores(), one can check on the number of available cores on the local machine.

At the moment, the starting values are not chosen in a terribly clever way and could be far off. As a result, the optimizer may be slow to converge or may even get stuck at a local maximum. One can set the starting values manually for the various variance components and correlations in the model via the control argument by specifying the vectors sigma2.init, tau2.init, rho.init, gamma2.init, and/or phi.init as needed. Especially for complex models, it is a good idea to try out different starting values to make sure that the same estimates are obtained.

Information on the progress of the optimization algorithm can be obtained by setting verbose=TRUE (this won't work when using parallelization). Since fitting complex models with many random effects can be computationally expensive, this option is useful to determine how the model fitting is progressing. One can also set verbose to an integer (verbose=2 yields even more information and verbose=3 also sets option(warn=1) temporarily).

Whether particular variance components and/or correlations are actually identifiable needs to be carefully examined when fitting complex models. The function does some limited checking internally to fix variances and/or correlations at zero when it is clear that insufficient information is available to estimate a particular parameter (e.g., if a particular factor has only a single level, the corresponding variance component cannot be estimated). However, it is strongly advised in general to do post model fitting checks to make sure that the likelihood surface around the ML/REML estimates is not flat for some combination of the parameter estimates (which would imply that the estimates are essentially arbitrary). For example, one can plot the (restricted) log-likelihood as a function of each variance component and correlation in the model to make sure that each profile plot shows a clear peak at the corresponding ML/REML estimates. The profile.rma.mv function can be used for this purpose.

Finally, note that the model fitting is not done in a very efficient manner at the moment, which is partly a result of allowing for crossed random effects and correlations across the entire dataset (e.g., when using the R argument). As a result, the function works directly with the entire $k \times k$ (marginal) variance-covariance matrix of the observed effect sizes or outcomes (instead of working with smaller blocks in a block diagonal structure). As a result, model fitting can be slow for large k. However, when the variance-covariance structure is actually sparse, a lot of speed can be gained by setting sparse=TRUE, in which case sparse matrix objects are used (via the Matrix package). Also, when model fitting appears to be slow, setting verbose=TRUE is useful to obtain information on how the model fitting is progressing.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Berkey, C. S., Hoaglin, D. C., Antczak-Bouckoms, A., Mosteller, F., & Colditz, G. A. (1998). Meta-analysis of multiple outcomes by regression with random effects. *Statistics in Medicine*, **17**(22), 2537–2550. https://doi.org/10.1002/(sici)1097-0258(19981130)17:22<2537::aid-sim953>3.0.co;2-c

Gleser, L. J., & Olkin, I. (2009). Stochastically dependent effect sizes. In H. Cooper, L. V. Hedges, & J. C. Valentine (Eds.), *The handbook of research synthesis and meta-analysis* (2nd ed., pp. 357–376). New York: Russell Sage Foundation.

van Houwelingen, H. C., Arends, L. R., & Stijnen, T. (2002). Advanced methods in meta-analysis: Multivariate approach and meta-regression. *Statistics in Medicine*, **21**(4), 589–624. https://doi.org/10.1002/sim.1040

Ishak, K. J., Platt, R. W., Joseph, L., Hanley, J. A., & Caro, J. J. (2007). Meta-analysis of longitudinal studies. *Clinical Trials*, **4**(5), 525–539. https://doi.org/10.1177/1740774507083567

Kalaian, H. A., & Raudenbush, S. W. (1996). A multivariate mixed linear model for meta-analysis. *Psychological Methods*, **1**(3), 227-235. https://doi.org/10.1037/1082-989X.1.3.227

Konstantopoulos, S. (2011). Fixed effects and variance components estimation in three-level meta-analysis. *Research Synthesis Methods*, **2**(1), 61–76. https://doi.org/10.1002/jrsm.35

Nakagawa, S., & Santos, E. S. A. (2012). Methodological issues and advances in biological meta-analysis. *Evolutionary Ecology*, **26**(5), 1253–1274. https://doi.org/10.1007/s10682-012-9555-5

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

Steiger, J. H. (1980). Tests for comparing elements of a correlation matrix. *Psychological Bulletin*, **87**(2), 245–251. https://doi.org/10.1037/0033-2909.87.2.245

Salanti, G., Higgins, J. P. T., Ades, A. E., & Ioannidis, J. P. A. (2008). Evaluation of networks of randomized trials. *Statistical Methods in Medical Research*, **17**(3), 279–301. https://doi.org/10.1177/0962280207080643

Trikalinos, T. A., & Olkin, I. (2012). Meta-analysis of effect sizes reported at multiple time points: A multivariate approach. *Clinical Trials*, **9**(5), 610–620. https://doi.org/10.1177/1740774512453218

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mh, rma.peto, and rma.glmm for other model fitting functions.

dat.konstantopoulos2011, dat.hasselblad1998, dat.begg1989, dat.berkey1998, dat.fine1993, and dat.ishak2007 for further examples of the use of the rma.mv function.

Examples

```
### calculate log odds ratios and corresponding sampling variances
dat <- escalc(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit random-effects model using rma.uni()
rma(yi, vi, data=dat)
```

```
### fit random-effects model using rma.mv()
### note: sigma^2 in this model is the same as tau^2 from the previous model
rma.mv(yi, vi, random = ~ 1 | trial, data=dat)

### change data into long format
dat.long <- to.long(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### set levels of group variable ("exp" = experimental/vaccinated; "con" = control/non-vaccinated)
levels(dat.long$group) <- c("exp", "con")

### set "con" to reference level
dat.long$group <- relevel(dat.long$group, ref="con")

### calculate log odds and corresponding sampling variances
dat.long <- escalc(measure="PLO", xi=out1, mi=out2, data=dat.long)

### fit bivariate random-effects model using rma.mv()
res <- rma.mv(yi, vi, mods = ~ group, random = ~ group | study, struct="UN", data=dat.long)
res
```

rma.peto *Meta-Analysis via Peto's Method*

Description

Function to fit fixed-effects models to 2×2 table data via Peto's method. See below and the documentation of the **metafor-package** for more details on these models.

Usage

```
rma.peto(ai, bi, ci, di, n1i, n2i,
         data, slab, subset,
         add=1/2, to="only0", drop00=TRUE,
         level=95, digits, verbose=FALSE, ...)
```

Arguments

ai	vector to specify the 2×2 table frequencies (upper left cell). See below and the documentation of the escalc function for more details.
bi	vector to specify the 2×2 table frequencies (upper right cell). See below and the documentation of the escalc function for more details.
ci	vector to specify the 2×2 table frequencies (lower left cell). See below and the documentation of the escalc function for more details.
di	vector to specify the 2×2 table frequencies (lower right cell). See below and the documentation of the escalc function for more details.
n1i	vector to specify the group sizes or row totals (first group). See below and the documentation of the escalc function for more details.

`n2i` vector to specify the group sizes or row totals (second group). See below and the documentation of the `escalc` function for more details.

`data` optional data frame containing the data supplied to the function.

`slab` optional vector with labels for the k studies.

`subset` optional (logical or numeric) vector to specify the subset of studies that should be used for the analysis.

`add` non-negative number to specify the amount to add to zero cells, counts, or frequencies when calculating the observed effect sizes or outcomes of the individual studies. Can also be a vector of two numbers, where the first number is used in the calculation of the observed effect sizes outcomes and the second number is used when applying Peto's method. See below and the documentation of the `escalc` function for more details.

`to` character string to specify when the values under `add` should be added (either `"only0"`, `"all"`, `"if0all"`, or `"none"`). Can also be a character vector, where the first string again applies when calculating the observed effect sizes or outcomes and the second string when applying Peto's method. See below and the documentation of the `escalc` function for more details.

`drop00` logical to specify whether studies with no cases (or only cases) in both groups should be dropped when calculating the observed effect sizes or outcomes (the outcomes for such studies are set to `NA`). Can also be a vector of two logicals, where the first applies to the calculation of the observed effect sizes or outcomes and the second when applying Peto's method. See below and the documentation of the `escalc` function for more details.

`level` numeric value between 0 and 100 to specify the confidence interval level (the default is 95).

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is 4.

`verbose` logical to specify whether output should be generated on the progress of the model fitting (the default is `FALSE`).

`...` additional arguments.

Details

Specifying the Data

The studies are assumed to provide data in terms of 2×2 tables of the form:

	outcome 1	outcome 2	total
group 1	`ai`	`bi`	`n1i`
group 2	`ci`	`di`	`n2i`

where `ai`, `bi`, `ci`, and `di` denote the cell frequencies and `n1i` and `n2i` the row totals. For example, in a set of randomized clinical trials (RCTs) or cohort studies, group 1 and group 2 may refer to the treatment (exposed) and placebo/control (not exposed) group, with outcome 1 denoting some event of interest (e.g., death) and outcome 2 its complement. In a set of case-control studies, group 1 and group 2 may refer to the group of cases and the group of controls, with outcome 1 denoting, for example, exposure to some risk factor and outcome 2 non-exposure.

Peto's Method

An approach for aggregating data of this type was suggested by Peto (see Yusuf et al., 1985). The method provides a weighted estimate of the (log) odds ratio under a fixed-effects model. The method is particularly advantageous when the event of interest is rare, but it should only be used when the group sizes within the individual studies are not too dissimilar and the effect sizes are generally small (Greenland & Salvan, 1990; Sweeting et al., 2004; Bradburn et al., 2007). Note that the printed results are given both in terms of the log and the raw units (for easier interpretation).

Observed Effect Sizes or Outcomes of the Individual Studies

Peto's method itself does not require the calculation of the observed (log) odds ratios of the individual studies and directly makes use of the 2×2 table counts. Zero cells are not a problem (except in extreme cases, such as when one of the two outcomes never occurs in any of the tables). Therefore, it is unnecessary to add some constant to the cell counts when there are zero cells.

However, for plotting and various other functions, it is necessary to calculate the observed (log) odds ratios for the k studies. Here, zero cells can be problematic, so adding a constant value to the cell counts ensures that all k values can be calculated. The `add` and `to` arguments are used to specify what value should be added to the cell frequencies and under what circumstances when calculating the observed (log) odds ratios and when applying Peto's method. Similarly, the `drop00` argument is used to specify how studies with no cases (or only cases) in both groups should be handled. The documentation of the `escalc` function explains how the `add`, `to`, and `drop00` arguments work. If only a single value for these arguments is specified (as per default), then these values are used when calculating the observed (log) odds ratios and no adjustment to the cell counts is made when applying Peto's method. Alternatively, when specifying two values for these arguments, the first value applies when calculating the observed (log) odds ratios and the second value when applying Peto's method.

Note that `drop00` is set to `TRUE` by default. Therefore, the observed (log) odds ratios for studies where `ai=ci=0` or `bi=di=0` are set to `NA`. When applying Peto's method, such studies are not explicitly dropped (unless the second value of `drop00` argument is also set to `TRUE`), but this is practically not necessary, as they do not actually influence the results (assuming no adjustment to the cell/event counts are made when applying Peto's method).

Value

An object of class `c("rma.peto","rma")`. The object is a list containing the following components:

`beta`	aggregated log odds ratio.
`se`	standard error of the aggregated value.
`zval`	test statistics of the aggregated value.
`pval`	corresponding p-value.
`ci.lb`	lower bound of the confidence interval.
`ci.ub`	upper bound of the confidence interval.
`QE`	test statistic of the test for heterogeneity.
`QEp`	corresponding p-value.
`k`	number of studies included in the analysis.

`yi, vi` the vector of individual log odds ratios and corresponding sampling variances.

`fit.stats` a list with the log-likelihood, deviance, AIC, BIC, and AICc values under the unrestricted and restricted likelihood.

`...` some additional elements/values.

Methods

The results of the fitted model are formatted and printed with the `print.rma.peto` function. If fit statistics should also be given, use `summary.rma` (or use the `fitstats.rma` function to extract them).

The `residuals.rma`, `rstandard.rma.peto`, and `rstudent.rma.peto` functions extract raw and standardized residuals. Leave-one-out diagnostics can be obtained with `leave1out.rma.peto`.

Forest, funnel, radial, L'Abbé, and Baujat plots can be obtained with `forest.rma`, `funnel.rma`, `radial.rma`, `labbe.rma`, and `baujat.rma`. The `qqnorm.rma.peto` function provides normal QQ plots of the standardized residuals. One can also just call `plot.rma.peto` on the fitted model object to obtain various plots at once.

A cumulative meta-analysis (i.e., adding one observation at a time) can be obtained with `cumul.rma.peto`.

Other extractor functions include `coef.rma`, `vcov.rma`, `logLik.rma`, `deviance.rma`, `AIC.rma`, and `BIC.rma`.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Bradburn, M. J., Deeks, J. J., Berlin, J. A., & Localio, A. R. (2007). Much ado about nothing: A comparison of the performance of meta-analytical methods with rare events. *Statistics in Medicine*, **26**(1), 53–77. https://doi.org/10.1002/sim.2528

Greenland, S., & Salvan, A. (1990). Bias in the one-step method for pooling study results. *Statistics in Medicine*, **9**(3), 247–252. https://doi.org/10.1002/sim.4780090307

Sweeting, M. J., Sutton, A. J., & Lambert, P. C. (2004). What to add to nothing? Use and avoidance of continuity corrections in meta-analysis of sparse data. *Statistics in Medicine*, **23**(9), 1351–1375. https://doi.org/10.1002/sim.1761

Yusuf, S., Peto, R., Lewis, J., Collins, R., & Sleight, P. (1985). Beta blockade during and after myocardial infarction: An overview of the randomized trials. *Progress in Cardiovascular Disease*, **27**(5), 335–371. https://doi.org/10.1016/s0033-0620(85)80003-7

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`rma.uni`, `rma.glmm`, `rma.mh`, and `rma.mv` for other model fitting functions.

`dat.collins1985a`, `dat.collins1985b`, and `dat.yusuf1985` for further examples of the use of the `rma.peto` function.

Examples

```
### meta-analysis of the (log) odds ratios using Peto's method
rma.peto(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)
```

rma.uni *Meta-Analysis via Linear (Mixed-Effects) Models*

Description

Function to fit the meta-analytic fixed- and random/mixed-effects models with or without moderators via linear (mixed-effects) models. See the documentation of the **metafor-package** for more details on these models.

Usage

```
rma.uni(yi, vi, sei, weights, ai, bi, ci, di, n1i, n2i, x1i, x2i, t1i, t2i,
        m1i, m2i, sd1i, sd2i, xi, mi, ri, ti, sdi, r2i, ni, mods, scale,
        measure=”GEN”, intercept=TRUE, data, slab, subset,
        add=1/2, to=”only0”, drop00=FALSE, vtype=”LS”,
        method=”REML”, weighted=TRUE, test=”z”,
        level=95, digits, btt, att, tau2, verbose=FALSE, control, ...)
rma(yi, vi, sei, weights, ai, bi, ci, di, n1i, n2i, x1i, x2i, t1i, t2i,
        m1i, m2i, sd1i, sd2i, xi, mi, ri, ti, sdi, r2i, ni, mods, scale,
        measure=”GEN”, intercept=TRUE, data, slab, subset,
        add=1/2, to=”only0”, drop00=FALSE, vtype=”LS”,
        method=”REML”, weighted=TRUE, test=”z”,
        level=95, digits, btt, att, tau2, verbose=FALSE, control, ...)
```

Arguments

`yi` vector of length k with the observed effect sizes or outcomes. See ‘Details’.

`vi` vector of length k with the corresponding sampling variances. See ‘Details’.

`sei` vector of length k with the corresponding standard errors (only relevant when not using `vi`). See ‘Details’.

`weights` optional argument to specify a vector of length k with user-defined weights. See ‘Details’.

`ai` see below and the documentation of the `escalc` function for more details.

`bi` see below and the documentation of the `escalc` function for more details.

`ci` see below and the documentation of the `escalc` function for more details.

`di` see below and the documentation of the `escalc` function for more details.

`n1i` see below and the documentation of the `escalc` function for more details.

`n2i` see below and the documentation of the `escalc` function for more details.

`x1i` see below and the documentation of the `escalc` function for more details.

x2i see below and the documentation of the escalc function for more details.

t1i see below and the documentation of the escalc function for more details.

t2i see below and the documentation of the escalc function for more details.

m1i see below and the documentation of the escalc function for more details.

m2i see below and the documentation of the escalc function for more details.

sd1i see below and the documentation of the escalc function for more details.

sd2i see below and the documentation of the escalc function for more details.

xi see below and the documentation of the escalc function for more details.

mi see below and the documentation of the escalc function for more details.

ri see below and the documentation of the escalc function for more details.

ti see below and the documentation of the escalc function for more details.

sdi see below and the documentation of the escalc function for more details.

r2i see below and the documentation of the escalc function for more details.

ni see below and the documentation of the escalc function for more details.

mods optional argument to include one or more moderators in the model. A single moderator can be given as a vector of length k specifying the values of the moderator. Multiple moderators are specified by giving a matrix with k rows and as many columns as there are moderator variables. Alternatively, a model formula can be used to specify the model. See 'Details'.

scale optional argument to include one or more predictors for the scale part in a location-scale model. See 'Details'.

measure character string to specify the type of data supplied to the function. When measure="GEN" (default), the observed effect sizes or outcomes and corresponding sampling variances (or standard errors) should be supplied to the function via the yi, vi, and sei arguments (only one of the two, vi or sei, needs to be specified). Alternatively, one can set measure to one of the effect size or outcome measures described under the documentation for the escalc function in which case one must specify the required data via the appropriate arguments.

intercept logical to specify whether an intercept should be added to the model (the default is TRUE). Ignored when mods is a formula.

data optional data frame containing the data supplied to the function.

slab optional vector with labels for the k studies.

subset optional (logical or numeric) vector to specify the subset of studies that should be used for the analysis.

add see the documentation of the escalc function.

to see the documentation of the escalc function.

drop00 see the documentation of the escalc function.

vtype see the documentation of the escalc function.

method character string to specify whether a fixed- or a random/mixed-effects model should be fitted. A fixed-effects model (with or without moderators) is fitted when using method="FE". Random/mixed-effects models are fitted by setting method equal to one of the following: "DL", "HE", "SJ", "ML", "REML", "EB", "HS", "HSk", or "GENQ". Default is "REML". See 'Details'.

`weighted` logical to specify whether weighted (default) or unweighted estimation should be used to fit the model.

`test` character string to specify how test statistics and confidence intervals for the fixed effects should be computed. By default (`test="z"`), Wald-type tests and CIs are obtained, which are based on a standard normal distribution. When `test="t"`, a t-distribution is used instead. When `test="knha"`, the method by Knapp and Hartung (2003) is used. See 'Details'.

`level` numeric value between 0 and 100 to specify the confidence interval level (the default is 95).

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is 4.

`btt` optional vector of indices to specify which coefficients to include in the omnibus test of moderators. Can also be a string to grep for. See 'Details'.

`att` optional vector of indices to specify which scale coefficients to include in the omnibus test. Only relevant for location-scale models. See 'Details'.

`tau2` optional numeric value to specify the amount of (residual) heterogeneity in a random- or mixed-effects model (instead of estimating it). Useful for sensitivity analyses (e.g., for plotting results as a function of τ^2). When unspecified, the value of τ^2 is estimated from the data.

`verbose` logical to specify whether output should be generated on the progress of the model fitting (the default is `FALSE`). Can also be an integer. Values > 1 generate more verbose output. See 'Note'.

`control` optional list of control values for the iterative estimation algorithms. If unspecified, default values are defined inside the function. See 'Note'.

`...` additional arguments.

Details

Specifying the Data

The function can be used in conjunction with any of the usual effect size or outcome measures used in meta-analyses (e.g., log risk ratios, log odds ratios, risk differences, mean differences, standardized mean differences, log transformed ratios of means, raw correlation coefficients, correlation coefficients transformed with Fisher's r-to-z transformation, and so on). Simply specify the observed effect sizes or outcomes via the `yi` argument and the corresponding sampling variances via the `vi` argument. Instead of specifying `vi`, one can specify the standard errors (the square root of the sampling variances) via the `sei` argument. The `escalc` function can be used to compute a wide variety of effect size or outcome measures (and the corresponding sampling variances) based on summary statistics.

Alternatively, the function can automatically calculate the values of a chosen effect size or outcome measure (and the corresponding sampling variances) when supplied with the necessary data. The `escalc` function describes which effect size or outcome measures are currently implemented and what data/arguments should then be specified/used. The `measure` argument should then be set to the desired effect size or outcome measure.

Specifying the Model

The function can be used to fit fixed- and random/mixed-effects models, as well as meta-regression models including moderators (the difference between the various models is described in detail on the introductory **metafor-package** help page).

Assuming the observed effect sizes or outcomes and corresponding sampling variances are supplied via `yi` and `vi`, a *fixed-effects model* can be fitted with `rma(yi,vi,method="FE")`. Weighted estimation (with inverse-variance weights) is used by default. User-defined weights can be supplied via the `weights` argument. Unweighted estimation can be used by setting `weighted=FALSE` (which is the same as setting the weights equal to a constant).

A *random-effects model* can be fitted with the same code but setting the `method` argument to one of the various estimators for the amount of heterogeneity:

- `method="DL"` = DerSimonian-Laird estimator,
- `method="HE"` = Hedges estimator,
- `method="HS"` = Hunter-Schmidt estimator,
- `method="HSk"` = Hunter-Schmidt estimator with a small sample-size correction,
- `method="SJ"` = Sidik-Jonkman estimator,
- `method="ML"` = maximum-likelihood estimator,
- `method="REML"` = restricted maximum-likelihood estimator,
- `method="EB"` = empirical Bayes estimator,
- `method="PM"` = Paule-Mandel estimator,
- `method="GENQ"` = generalized Q-statistic estimator.

For a description of the various estimators, see Brannick et al. (2019), DerSimonian and Kacker (2007), Raudenbush (2009), Viechtbauer (2005), and Viechtbauer et al. (2015). Note that the Hedges estimator is also called the ‘variance component estimator’ or ‘Cochran estimator’, the Sidik-Jonkman estimator is also called the ‘model error variance estimator’, and the empirical Bayes estimator is actually identical to the Paule-Mandel estimator (Paule & Mandel, 1982). Finally, the generalized Q-statistic estimator is a general method-of-moments estimator (DerSimonian & Kacker, 2007) requiring the specification of weights (the HE and DL estimators are just special cases with equal and inverse variance weights, respectively).

One or more moderators can be included in these models via the `mods` argument. A single moderator can be given as a (row or column) vector of length k specifying the values of the moderator. Multiple moderators are specified by giving an appropriate model matrix (i.e., X) with k rows and as many columns as there are moderator variables (e.g., `mods = cbind(mod1,mod2,mod3)`, where `mod1`, `mod2`, and `mod3` correspond to the names of the variables for three moderator variables). The intercept is added to the model matrix by default unless `intercept=FALSE`.

Alternatively, one can use standard `formula` syntax to specify the model. In this case, the `mods` argument should be set equal to a one-sided formula of the form `mods = ~ model` (e.g., `mods = ~ mod1 + mod2 + mod3`). Interactions, polynomial terms, and factors can be easily added to the model in this manner. When specifying a model formula via the `mods` argument, the `intercept` argument is ignored. Instead, the inclusion/exclusion of the intercept is controlled by the specified formula (e.g., `mods = ~ mod1 + mod2 + mod3 -1` would lead to the removal of the intercept).

When the observed effect sizes or outcomes and corresponding sampling variances are supplied via the `yi` and `vi` (or `sei`) arguments, one can also specify moderators via the `yi` argument (e.g., `rma(yi`

~ mod1 + mod2 + mod3,vi)). In that case, the mods argument is ignored and the inclusion/exclusion of the intercept again is controlled by the specified formula.

Omnibus Test of Moderators

For models including moderators, an omnibus test of all model coefficients is conducted that excludes the intercept (the first coefficient) if it is included in the model. If no intercept is included in the model, then the omnibus test includes all of the coefficients in the model including the first. Alternatively, one can manually specify the indices of the coefficients to test via the btt argument. For example, with btt=c(3,4), only the third and fourth coefficient from the model would be included in the test (if an intercept is included in the model, then it corresponds to the first coefficient in the model). Instead of specifying the coefficient numbers, one can specify a string for btt. In that case, grep will be used to search for all coefficient names that match the string. The omnibus test is called the Q_M-test and follows, under the assumptions of the model, a chi-square distribution with m degrees of freedom (with m denoting the number of coefficients tested) under the null hypothesis (that the true value of all coefficients tested is equal to 0).

Categorical Moderators

Categorical moderator variables can be included in the model via the mods argument in the same way that appropriately (dummy) coded categorical independent variables can be included in linear models. One can either do the dummy coding manually or use a model formula together with the factor function to let R handle the coding automatically (note that string/character variables in a model formula are automatically converted to factors). An example to illustrate these different approaches is provided below.

Tests and Confidence Intervals

By default, tests of individual coefficients in the model (and the corresponding confidence intervals) are based on a standard normal distribution, while the omnibus test is based on a chi-square distribution (see above). As an alternative, one can set test="t", in which case tests of individual coefficients and confidence intervals are based on a t-distribution with $k - p$ degrees of freedom, while the omnibus test statistic then uses an F-distribution with m and $k-p$ degrees of freedom (with k denoting the total number of estimates included in the analysis and p the total number of model coefficients including the intercept if it is present). Finally, when test="knha", the Knapp and Hartung (2003) method is used, which applies an adjustment to the standard errors of the estimated coefficients (to account for the uncertainty in the estimate of the amount of (residual) heterogeneity) and uses t- and F-distributions as described above.

Test for (Residual) Heterogeneity

A test for (residual) heterogeneity is automatically carried out by the function. Without moderators in the model, this is simply Cochran's Q-test (Cochran, 1954), which tests whether the variability in the observed effect sizes or outcomes is larger than would be expected based on sampling variability alone. A significant test suggests that the true effects/outcomes are heterogeneous. When moderators are included in the model, this is the Q_E-test for residual heterogeneity, which tests whether the variability in the observed effect sizes or outcomes not accounted for by the moderators included in the model is larger than would be expected based on sampling variability alone.

Location-Scale Models

The function can also be used to fit so-called 'location-scale models'. In such models, one can specify not only predictors for the size of the average true outcome (i.e., for their 'location'), but also predictors for the amount of heterogeneity in the outcomes (i.e., their 'scale'). The model is then given by

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_{p'} x_{ip'} + u_i + \epsilon_i,$$

$$u_i \sim N(0, \tau_i^2), \; \epsilon_i \sim N(0, v_i),$$
$$\ln(\tau_i^2) = \alpha_0 + \alpha_1 z_{i1} + \alpha_2 z_{i2} + \ldots + \alpha_{q'} z_{iq'},$$

where $x_{i1}, \ldots, x_{ip'}$ are the values of the p' predictor variables that may be related to the size of the average true outcome (letting $p = p' + 1$ denote the total number of location coefficients in the model including the model intercept β_0) and $z_{i1}, \ldots, z_{iq'}$ are the values of the q' scale variables that may be related to the amount of heterogeneity in the outcomes (letting $q = q' + 1$ denote the total number of scale coefficients in the model including the model intercept α_0). Location variables can be specified via the `mods` argument as described above (e.g., `mods = ~ mod1 + mod2 + mod3`). Scale variables can be specified via the `scale` argument (e.g., `scale = ~ var1 + var2 + var3`). A log link is used for specifying the relationship between the scale variables and the amount of heterogeneity so that τ_i^2 is guaranteed to be non-negative. Estimates of the location and scale coefficients can be obtained with either maximum likelihood (`method="ML"`) or restricted maximum likelihood (`method="REML"`) estimation. An omnibus test of the scale coefficients is conducted as described above (where the `att` argument can be used to specify which coefficients to include in the test).

Value

An object of class `c("rma.uni","rma")`. The object is a list containing the following components:

`beta` estimated coefficients of the model.

`se` standard errors of the coefficients.

`zval` test statistics of the coefficients.

`pval` corresponding p-values.

`ci.lb` lower bound of the confidence intervals for the coefficients.

`ci.ub` upper bound of the confidence intervals for the coefficients.

`vb` variance-covariance matrix of the estimated coefficients.

`tau2` estimated amount of (residual) heterogeneity. Always `0` when `method="FE"`.

`se.tau2` standard error of the estimated amount of (residual) heterogeneity.

`k` number of studies included in the analysis.

`p` number of coefficients in the model (including the intercept).

`m` number of coefficients included in the omnibus test of moderators.

`QE` test statistic of the test for (residual) heterogeneity.

`QEp` corresponding p-value.

`QM` test statistic of the omnibus test of moderators.

`QMp` corresponding p-value.

`I2` value of I^2. See `print.rma.uni` for more details.

`H2` value of H^2. See `print.rma.uni` for more details.

`R2` value of R^2. See `print.rma.uni` for more details.

`int.only` logical that indicates whether the model is an intercept-only model.

`yi, vi, X` the vector of outcomes, the corresponding sampling variances, and the model matrix.

`fit.stats` a list with the log-likelihood, deviance, AIC, BIC, and AICc values under the unrestricted and restricted likelihood.

`...` some additional elements/values.

For location-scale models, the object is of class `c("rma.ls","rma.uni","rma")` and includes the following components in addition to the ones listed above:

`alpha` estimated scale coefficients of the model.

`se.alpha` standard errors of the coefficients.

`zval.alpha` test statistics of the coefficients.

`pval.alpha` corresponding p-values.

`ci.lb.alpha` lower bound of the confidence intervals for the coefficients.

`ci.ub.alpha` upper bound of the confidence intervals for the coefficients.

`va` variance-covariance matrix of the estimated coefficients.

`tau2` as above, but now a vector of values.

`q` number of scale coefficients in the model (including the intercept).

`QS` test statistic of the omnibus test of the scale coefficients.

`QSp` corresponding p-value.

`...` some additional elements/values.

Methods

The results of the fitted model are formatted and printed with the `print.rma.uni` function. If fit statistics should also be given, use `summary.rma` (or use the `fitstats.rma` function to extract them). Full versus reduced model comparisons in terms of fit statistics and likelihoods can be obtained with `anova.rma`. Wald-type tests for sets of model coefficients or linear combinations thereof can be obtained with the same function. Permutation tests for the model coefficient(s) can be obtained with `permutest.rma.uni`. Tests and confidence intervals based on (cluster) robust methods can be obtained with `robust.rma.uni`.

Predicted/fitted values can be obtained with `predict.rma` and `fitted.rma`. For best linear unbiased predictions, see `blup.rma.uni` and `ranef.rma.uni`.

The `residuals.rma`, `rstandard.rma.uni`, and `rstudent.rma.uni` functions extract raw and standardized residuals. Additional case diagnostics (e.g., to determine influential studies) can be obtained with the `influence.rma.uni` function. For models without moderators, leave-one-out diagnostics can also be obtained with `leave1out.rma.uni`. For models with moderators, variance inflation factors can be obtained with `vif.rma`.

A confidence interval for the amount of (residual) heterogeneity in the random/mixed-effects model can be obtained with `confint.rma.uni`.

Forest, funnel, radial, L'Abbé, and Baujat plots can be obtained with `forest.rma`, `funnel.rma`, `radial.rma`, `labbe.rma`, and `baujat.rma` (radial and L'Abbé plots only for models without moderators). The `qqnorm.rma.uni` function provides normal QQ plots of the standardized residuals. One can also just call `plot.rma.uni` on the fitted model object to obtain various plots at once. For random/mixed-effects models, the `profile.rma.uni` function can be used to obtain a plot of the (restricted) log-likelihood as a function of τ^2. For models with moderators, `regplot.rma` draws scatter plots / bubble plots, showing the (marginal) relationship between the observed outcomes and a selected moderator from the model.

Tests for funnel plot asymmetry (which may be indicative of publication bias) can be obtained with `ranktest.rma` and `regtest.rma`. For models without moderators, the `trimfill.rma.uni` method can be used to carry out a trim and fill analysis and `hc.rma.uni` provides a random-effects model analysis that is more robust to publication bias (based on the method by Henmi & Copas, 2010). The test of 'excess significance' can be carried out with the `tes.rma` function. Selection models can be fitted with the `selmodel` function.

For models without moderators, a cumulative meta-analysis (i.e., adding one observation at a time) can be obtained with `cumul.rma.uni`.

Other extractor functions include `coef.rma`, `vcov.rma`, `logLik.rma`, `deviance.rma`, `AIC.rma`, `BIC.rma`, `hatvalues.rma.uni`, and `weights.rma.uni`.

Note

While the HS, HSk, HE, DL, SJ, and GENQ estimators of τ^2 are based on closed-form solutions, the ML, REML, and EB estimators must be obtained iteratively. For this, the function makes use of the Fisher scoring algorithm, which is robust to poor starting values and usually converges quickly (Harville, 1977; Jennrich & Sampson, 1976). By default, the starting value is set equal to the value of the Hedges (HE) estimator and the algorithm terminates when the change in the estimated value of τ^2 is smaller than 10^{-5} from one iteration to the next. The maximum number of iterations is 100 by default (which should be sufficient in most cases). Information on the progress of the algorithm can be obtained by setting `verbose=TRUE`. One can also set `verbose` to an integer (`verbose=2` yields even more information and `verbose=3` also sets `option(warn=1)` temporarily).

A different starting value, threshold, and maximum number of iterations can be specified via the `control` argument by setting `control=list(tau2.init=value,threshold=value,maxiter=value)`. The step length of the Fisher scoring algorithm can also be adjusted by a desired factor with `control=list(stepadj=value)` (values below 1 will reduce the step length). If using `verbose=TRUE` shows the estimate jumping around erratically (or cycling through a few values), decreasing the step length (and increasing the maximum number of iterations) can often help with convergence (e.g., `control=list(stepadj=0.5,maxiter=1000)`).

The PM estimator also involves an iterative algorithm, which makes use of the `uniroot` function. By default, the desired accuracy (`tol`) is set equal to `.Machine$double.eps^0.25` and the maximum number of iterations (`maxiter`) to `100` (as above). The upper bound of the interval searched (`tau2.max`) is set to 100 (which should be large enough for most cases). These values can be adjusted with `control=list(tol=value,maxiter=value,tau2.max=value)`.

All of the heterogeneity estimators except SJ can in principle yield negative estimates for the amount of (residual) heterogeneity. However, negative estimates of τ^2 are outside of the parameter space. For the HS, HSk, HE, DL, and GENQ estimators, negative estimates are therefore truncated to zero. For the ML, REML, and EB estimators, the Fisher scoring algorithm makes use of step halving (Jennrich & Sampson, 1976) to guarantee a non-negative estimate. Finally, for the PM estimator, the lower bound of the interval searched is set to zero by default. For those brave enough to step into risky territory, there is the option to set the lower bound for all these estimators to some other value besides zero (even a negative one) with `control=list(tau2.min=value)`, but the lowest value permitted is `-min(vi)` (to ensure that the marginal variances are always non-negative).

The Hunter-Schmidt estimator for the amount of heterogeneity is defined in Hunter and Schmidt (1990) only in the context of the random-effects model when analyzing correlation coefficients.

A general version of this estimator for random- and mixed-effects models not specific to any particular outcome measure is described in Viechtbauer (2005) and Viechtbauer et al. (2015) and is implemented here.

The Sidik-Jonkman estimator starts with a crude estimate of τ^2, which is then updated as described in Sidik and Jonkman (2005b, 2007). If, instead of the crude estimate, one wants to use a better a priori estimate, one can do so by passing this value via `control=list(tau2.init=value)`.

Outcomes with non-positive sampling variances are problematic. If a sampling variance is equal to zero, then its weight will be $1/0$ for fixed-effects models when using weighted estimation. Switching to unweighted estimation is a possible solution then. For random/mixed-effects model, some estimators of τ^2 are undefined when there is at least one sampling variance equal to zero. Other estimators may work, but it may still be necessary to switch to unweighted model fitting, especially when the estimate of τ^2 converges to zero.

When including moderators in the model, it is possible that the model matrix is not of full rank (i.e., there is a linear relationship between the moderator variables included in the model). The function automatically tries to reduce the model matrix to full rank by removing redundant predictors, but if this fails the model cannot be fitted and an error will be issued. Deleting (redundant) moderator variables from the model as needed should solve this problem.

Finally, some general words of caution about the assumptions underlying the models:

- The sampling variances (i.e., the `vi` values) are treated as if they are known constants. This (usually) implies that the distributions of the test statistics and corresponding confidence intervals are only exact and have nominal coverage when the within-study sample sizes are large (i.e., when the error in the sampling variance estimates is small). Certain outcome measures (e.g., the arcsine square root transformed risk difference and Fisher's r-to-z transformed correlation coefficient) are based on variance stabilizing transformations that also help to make the assumption of known sampling variances much more reasonable.
- When fitting a mixed/random-effects model, τ^2 is estimated and then treated as a known constant thereafter. This ignores the uncertainty in the estimate of τ^2. As a consequence, the standard errors of the parameter estimates tend to be too small, yielding test statistics that are too large and confidence intervals that are not wide enough. The Knapp and Hartung (2003) adjustment can be used to counter this problem, yielding test statistics and confidence intervals whose properties are closer to nominal.
- Most effect size or outcome measures are not exactly normally distributed as assumed under the various models. However, the normal approximation usually becomes more accurate for most effect size or outcome measures as the within-study sample sizes increase. Therefore, sufficiently large within-study sample sizes are (usually) needed to be certain that the tests and confidence intervals have nominal levels/coverage. Again, certain outcome measures (e.g., Fisher's r-to-z transformed correlation coefficient) may be preferable from this perspective as well.

Author(s)

Wolfgang Viechtbauer <`wvb@metafor-project.org`> `https://www.metafor-project.org`

References

Berkey, C. S., Hoaglin, D. C., Mosteller, F., & Colditz, G. A. (1995). A random-effects regression model for meta-analysis. *Statistics in Medicine*, **14**(4), 395–411. https://doi.org/10.1002/sim.4780140406

Brannick, M. T., Potter, S. M., Benitez, B., & Morris, S. B. (2019). Bias and precision of alternate estimators in meta-analysis: Benefits of blending Schmidt–Hunter and Hedges approaches. *Organizational Research Methods*, **22**(2), 490–514. https://doi.org/10.1177/1094428117741966

Cochran, W. G. (1954). The combination of estimates from different experiments. *Biometrics*, **10**(1), 101–129. https://doi.org/10.2307/3001666

DerSimonian, R., & Laird, N. (1986). Meta-analysis in clinical trials. *Controlled Clinical Trials*, **7**(3), 177–188. https://doi.org/10.1016/0197-2456(86)90046-2

DerSimonian, R., & Kacker, R. (2007). Random-effects model for meta-analysis of clinical trials: An update. *Contemporary Clinical Trials*, **28**(2), 105–114. https://doi.org/10.1016/j.cct.2006.04.004

Harville, D. A. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, **72**(358), 320–338. https://doi.org/10.2307/2286796

Hedges, L. V. (1983). A random effects model for effect sizes. *Psychological Bulletin*, **93**(2), 388–395. https://doi.org/10.1037/0033-2909.93.2.388

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. San Diego, CA: Academic Press.

Henmi, M., & Copas, J. B. (2010). Confidence intervals for random effects meta-analysis and robustness to publication bias. *Statistics in Medicine*, **29**(29), 2969–2983. https://doi.org/10.1002/sim.4029

Hunter, J. E., & Schmidt, F. L. (2004). *Methods of meta-analysis: Correcting error and bias in research findings* (2nd ed.). Thousand Oaks, CA: Sage.

Jennrich, R. I., & Sampson, P. F. (1976). Newton-Raphson and related algorithms for maximum likelihood variance component estimation. *Technometrics*, **18**(1), 11–17. https://doi.org/10.2307/1267911

Knapp, G., & Hartung, J. (2003). Improved tests for a random effects meta-regression with a single covariate. *Statistics in Medicine*, **22**(17), 2693–2710. https://doi.org/10.1002/sim.1482

Morris, C. N. (1983). Parametric empirical Bayes inference: Theory and applications. *Journal of the American Statistical Association*, **78**(381), 47–55. https://doi.org/10.2307/2287098

Paule, R. C., & Mandel, J. (1982). Consensus values and weighting factors. *Journal of Research of the National Bureau of Standards*, **87**(5), 377–385. https://doi.org/10.6028/jres.087.022

Raudenbush, S. W. (2009). Analyzing effect sizes: Random effects models. In H. Cooper, L. V. Hedges, & J. C. Valentine (Eds.), *The handbook of research synthesis and meta-analysis* (2nd ed., pp. 295–315). New York: Russell Sage Foundation.

Sidik, K., & Jonkman, J. N. (2005a). A note on variance estimation in random effects meta-regression. *Journal of Biopharmaceutical Statistics*, **15**(5), 823–838. https://doi.org/10.1081/BIP-200067915

Sidik, K., & Jonkman, J. N. (2005b). Simple heterogeneity variance estimation for meta-analysis. *Journal of the Royal Statistical Society, Series C*, **54**(2), 367–384. https://doi.org/10.1111/j.1467-9876.2005.00489.x

Sidik, K., & Jonkman, J. N. (2007). A comparison of heterogeneity variance estimators in combining results of studies. *Statistics in Medicine*, **26**(9), 1964–1981. https://doi.org/10.1002/sim.2688

Viechtbauer, W. (2005). Bias and efficiency of meta-analytic variance estimators in the random-effects model. *Journal of Educational and Behavioral Statistics*, **30**(3), 261–293. https://doi.org/10.3102/1076998603000326

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Viechtbauer, W., López-López, J. A., Sánchez-Meca, J., & Marín-Martínez, F. (2015). A comparison of procedures to test for moderators in mixed-effects meta-regression models. *Psychological Methods*, **20**(3), 360–374. https://doi.org/10.1037/met0000023

See Also

`rma.mh`, `rma.peto`, `rma.glmm`, and `rma.mv` for other model fitting functions.

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit a random-effects model using the log risk ratios and variances as input
### note: method="REML" is the default, so one could leave this out
rma(yi, vi, data=dat, method="REML")

### fit a random-effects model using the log risk ratios and standard errors as input
### note: the second argument of rma() is for the *variances*, so we use the
### named argument 'sei' to supply the standard errors to the function
dat$sei <- sqrt(dat$vi)
rma(yi, sei=sei, data=dat)

### fit a random-effects model supplying the 2x2 table cell frequencies to the function
rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat)

### fit a mixed-effects model with two moderators (absolute latitude and publication year)
rma(yi, vi, mods=cbind(ablat, year), data=dat)

### using a model formula to specify the same model
rma(yi, vi, mods = ~ ablat + year, data=dat)

### using a model formula as part of the yi argument
rma(yi ~ ablat + year, vi, data=dat)

### manual dummy coding of the allocation factor
alloc.random     <- ifelse(dat$alloc == "random",     1, 0)
alloc.alternate  <- ifelse(dat$alloc == "alternate",  1, 0)
alloc.systematic <- ifelse(dat$alloc == "systematic", 1, 0)

### test the allocation factor (in the presence of the other moderators)
### note: 'alternate' is the reference level of the allocation factor,
###       since this is the dummy/level we leave out of the model
### note: the intercept is the first coefficient, so with btt=2:3 we test
###       coefficients 2 and 3, corresponding to the coefficients for the
###       allocation factor
rma(yi, vi, mods = ~ alloc.random + alloc.systematic + year + ablat, data=dat, btt=2:3)

### using a model formula to specify the same model
rma(yi, vi, mods = ~ factor(alloc) + year + ablat, data=dat, btt=2:3)

### factor() is not needed as character variables are automatically converted to factors
```

```
rma(yi, vi, mods = ~ alloc + year + ablat, data=dat, btt=2:3)

### test all pairwise differences with Holm's method (using the 'multcomp' package if installed)
res <- rma(yi, vi, mods = ~ factor(alloc) - 1, data=dat)
res
if (require(multcomp))
   summary(glht(res, linfct=contrMat(c("alternate"=1,"random"=1,"systematic"=1),
           type="Tukey")), test=adjusted("holm"))

### subgrouping versus using a single model with a factor (subgrouping provides
### an estimate of tau^2 within each subgroup, but the number of studies in each
### subgroup is quite small; the model with the allocation factor provides a
### single estimate of tau^2 based on a larger number of studies, but assumes
### that tau^2 is the same within each subgroup)
res.a <- rma(yi, vi, data=dat, subset=(alloc=="alternate"))
res.r <- rma(yi, vi, data=dat, subset=(alloc=="random"))
res.s <- rma(yi, vi, data=dat, subset=(alloc=="systematic"))
res.a
res.r
res.s
res <- rma(yi, vi, mods = ~ factor(alloc) - 1, data=dat)
res

############################################################################

### demonstrating that Q_E + Q_M = Q_Total for fixed-effects models
### note: this does not work for random/mixed-effects models, since Q_E and
### Q_Total are calculated under the assumption that tau^2 = 0, while the
### calculation of Q_M incorporates the estimate of tau^2
res <- rma(yi, vi, data=dat, method="FE")
res ### this gives Q_Total
res <- rma(yi, vi, mods = ~ ablat + year, data=dat, method="FE")
res ### this gives Q_E and Q_M
res$QE + res$QM

### decomposition of Q_E into subgroup Q-values
res <- rma(yi, vi, mods = ~ factor(alloc), data=dat)
res

res.a <- rma(yi, vi, data=dat, subset=(alloc=="alternate"))
res.r <- rma(yi, vi, data=dat, subset=(alloc=="random"))
res.s <- rma(yi, vi, data=dat, subset=(alloc=="systematic"))

res.a$QE ### Q-value within subgroup "alternate"
res.r$QE ### Q-value within subgroup "random"
res.s$QE ### Q-value within subgroup "systematic"

res$QE
res.a$QE + res.r$QE + res.s$QE

############################################################################

### an example of a location-scale model
```

```
dat <- dat.bangertdrowns2004

### fit a standard random-effects model
res <- rma(yi, vi, data=dat)
res

### fit the same model as a location-scale model
res <- rma(yi, vi, scale = ~ 1, data=dat)
res

### check that we obtain the same estimate for tau^2
predict(res, newscale=1, transf=exp)

### add the total sample size (per 100) as a location and scale predictor
dat$ni100 <- dat$ni/100
res <- rma(yi, vi, mods = ~ ni100, scale = ~ ni100, data=dat)
res

### variables in the location and scale parts can differ
res <- rma(yi, vi, mods = ~ ni100 + meta, scale = ~ ni100 + imag, data=dat)
res
```

robust *(Cluster) Robust Tests and Confidence Intervals for 'rma' Objects*

Description

The function provides (cluster) robust tests and confidence intervals of the model coefficients for objects of class "rma".

Usage

```
robust(x, cluster, ...)

## S3 method for class 'rma.uni'
robust(x, cluster, adjust=TRUE, digits, ...)
## S3 method for class 'rma.mv'
robust(x, cluster, adjust=TRUE, digits, ...)
```

Arguments

x	an object of class "rma.uni" or "rma.mv".
cluster	a vector to specify a clustering variable to use for constructing the sandwich estimator of the variance-covariance matrix.
adjust	logical to specify whether a small-sample correction should be applied to the variance-covariance matrix.
digits	integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.

`...`	other arguments.

Details

The function constructs a (cluster) robust estimate of the variance-covariance matrix of the model coefficients based on a sandwich-type estimator and then computes tests and confidence intervals of the model coefficients.

Tests of individual coefficients and confidence intervals are based on a t-distribution with $n - p$ degrees of freedom is used, while the omnibus test statistic uses an F-distribution with m and $n - p$ degrees of freedom, where n is the number of clusters, p denotes the total number of model coefficients (including the intercept if it is present), and m denotes the number of coefficients tested (in the omnibus test).

When `adjust=TRUE` (the default), the (cluster) robust estimate of the variance-covariance matrix is multiplied by the factor $n/(n-p)$, which serves as a small-sample adjustment that tends to improve the performance of the method when the number of clusters is small.

For even better small-sample adjustments (Pustejovsky & Tipton, 2018), see the clubSandwich package, which nicely works together with the **metafor** package (see 'Examples').

Value

An object of class `"robust.rma"`. The object is a list containing the following components:

`beta`	estimated coefficients of the model.
`se`	robust standard errors of the coefficients.
`zval`	test statistics of the coefficients.
`pval`	corresponding p-values.
`ci.lb`	lower bound of the confidence intervals for the coefficients.
`ci.ub`	upper bound of the confidence intervals for the coefficients.
`vb`	robust variance-covariance matrix of the estimated coefficients.
`QM`	test statistic of the omnibus test of moderators.
`QMp`	corresponding p-value.
`...`	some additional elements/values.

The results are formatted and printed with the `print.robust.rma` function.

Note

The variable specified via `cluster` is assumed to be of the same length as the data originally passed to the `rma.uni` or `rma.mv` functions. Any subsetting and removal of studies with missing values that was applied during the model fitting is also automatically applied to the variable specified via the `cluster` argument.

The idea of the robust (sandwich-type) estimator for models with unspecified heteroscedasticity can be traced back to Eicker (1967), Huber (1967), and White (1980). Hence, the method in general is often referred to as the Eicker-Huber-White method. Some small-sample improvements to the method are described by MacKinnon and White (1985). The extension to the cluster robust estimator can be found in Froot (1989) and Williams (2000). Cameron and Miller (2015) provide an extensive overview of cluster robust methods. Sidik and Jonkman (2005, 2006) introduced robust methods in the meta-analytic context for standard random/mixed-effects models. The use of the cluster robust estimator for multivariate/multilevel meta-analytic models is described in Hedges, Tipton, and Johnson (2010).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Cameron, A. C., & Miller, D. L. (2015). A practitioner's guide to cluster-robust inference. *Journal of Human Resources*, **50**(2), 317–372. https://doi.org/10.3368/jhr.50.2.317

Eicker, F. (1967). Limit theorems for regressions with unequal and dependent errors. In L. M. LeCam & J. Neyman (Eds.), *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 59–82). Berkeley: University of California Press.

Froot, K. A. (1989). Consistent covariance matrix estimation with cross-sectional dependence and heteroskedasticity in financial data. *Journal of Financial and Quantitative Analysis*, **24**(3), 333–355. https://doi.org/10.2307/2330815

Hedges, L. V., Tipton, E., & Johnson, M. C. (2010). Robust variance estimation in meta-regression with dependent effect size estimates. *Research Synthesis Methods*, **1**(1), 39–65. https://doi.org/10.1002/jrsm.5

Huber, P. (1967). The behavior of maximum-likelihood estimates under nonstandard conditions. In L. M. LeCam & J. Neyman (Eds.), *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 221–233). Berkeley: University of California Press.

MacKinnon, J. G., & White, H. (1985). Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, **29**(3), 305–325. https://doi.org/10.1016/0304-4076(85)90158-7

Pustejovsky, J. E., & Tipton, E. (2018). Small-sample methods for cluster-robust variance estimation and hypothesis testing in fixed effects models. *Journal of Business & Economic Statistics*, **36**(4), 672–683. https://doi.org/10.1080/07350015.2016.1247004

Sidik, K., & Jonkman, J. N. (2005). A note on variance estimation in random effects meta-regression. *Journal of Biopharmaceutical Statistics*, **15**(5), 823–838. https://doi.org/10.1081/BIP-200067915

Sidik, K., & Jonkman, J. N. (2006). Robust variance estimation for random effects meta-analysis. *Computational Statistics & Data Analysis*, **50**(12), 3681–3701. https://doi.org/10.1016/j.csda.2005.07.019

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, **48**(4), 817–838. https://doi.org/10.2307/1912934

Williams, R. L. (2000). A note on robust variance estimation for cluster-correlated data. *Biometrics*, **56**(2), 645–646. https://doi.org/10.1111/j.0006-341x.2000.00645.x

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mv

Examples

```
### copy data from Konstantopoulos (2011) into 'dat'
dat <- dat.konstantopoulos2011

### fit multilevel random-effects model
res <- rma.mv(yi, vi, random = ~ 1 | district/school, data=dat)
res

### obtain results based on the sandwich method
robust(res, cluster=dat$district)

### illustrate use of the clubSandwich package together with metafor
## Not run:
require(clubSandwich)
coef_test(res, vcov="CR2", cluster=dat$district)

## End(Not run)

### copy data from Berkey et al. (1998) into 'dat'
dat <- dat.berkey1998

### construct list with the variance-covariance matrices of the observed outcomes for the studies
V <- lapply(split(dat[c("v1i", "v2i")], dat$trial), as.matrix)

### construct block diagonal matrix
V <- bldiag(V)

### fit multivariate model
res <- rma.mv(yi, V, mods = ~ outcome - 1, random = ~ outcome | trial, struct="UN", data=dat)
res

### obtain results based on sandwich method
robust(res, cluster=dat$trial)

### illustrate use of the clubSandwich package together with metafor
## Not run:
require(clubSandwich)
coef_test(res, vcov="CR2", cluster=dat$trial)

## End(Not run)
```

---

selmodel *Selection Models*

Description

Function to fit selection models.

Usage

```
selmodel(x, ...)

## S3 method for class 'rma.uni'
selmodel(x, type, alternative="greater", prec,
         delta, steps, verbose=FALSE, digits, control, ...)
```

Arguments

`x` an object of class `"rma.uni"`.

`type` character string to specify the type of selection model. Possible options are `"beta"`, `"halfnorm"`, `"negexp"`, `"logistic"`, `"power"`, `"negexppow"`, or `"stepfun"`. Can be abbreviated. See 'Details'.

`alternative` character string to specify the sidedness of the hypothesis when testing the observed outcomes. Possible options are `"greater"` (the default), `"less"`, or `"two.sided"`. Can be abbreviated.

`prec` optional character string to specify the measure of precision (only relevant for selection models that can incorporate this into the selection function). Possible options are `"sei"`, `"vi"`, `"ninv"`, or `"sqrtninv"`. See 'Details'.

`delta` optional numeric vector (of the same length as the number of selection model parameters) to fix the corresponding δ value(s). A specific δ value can be fixed by setting the corresponding element of this argument to the desired value. A specific δ value will be estimated if the corresponding element is set equal to `NA`. See 'Details'.

`steps` numeric vector of one or more values between 0 and 1 that can or must be specified for certain selection functions. See 'Details'.

`verbose` logical to specify whether output should be generated on the progress of the model fitting (the default is `FALSE`). Can also be an integer. Values > 1 generate more verbose output. See 'Note'.

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.

`control` optional list of control values for the estimation algorithm. If unspecified, default values are defined inside the function. See 'Note'.

`...` other arguments.

Details

Selection models are a general class of models that attempt to model the process by which the studies included in a meta-analysis may have been influenced by some form of publication bias. If a particular selection model is an adequate approximation for the underlying selection process, then the model provides estimates of the parameters of interest (e.g., the average true outcome and the amount of heterogeneity in the true outcomes) that are 'corrected' for this selection process (i.e., they are estimates of the parameters in the population of studies before any selection has taken place). The present function fits a variety of such selection models. To do so, one should pass an object fitted with the `rma.uni` function to the first argument. The model that will then be fitted is of the same form as the original model combined with the specific selection model chosen (see below for possible options). For example, if the original model was a random-effects model, then a random-effects selection model will be fitted. Similarly, if the original model included moderators, then they will also be included in the selection model. Model fitting is done via maximum likelihood (ML) estimation over the fixed- and random-effects parameters (e.g., μ and τ^2 in a random-effects model) and the selection model parameters.

Argument `type` determines the specific type of selection model that should be fitted. All selection models that can be fitted are based on the idea that selection may haven taken place based on the p-values of the studies. In particular, let y_i and v_i denote the observed outcome and the corresponding sampling variance of the ith study. Then $z_i = y_i/\sqrt{v_i}$ is the (Wald-type) test statistic for testing the null hypothesis H_0: $\theta_i = 0$ and $p_i = 1 - \Phi(z_i)$ (if `alternative="greater"`), $p_i = \Phi(z_i)$ (if `alternative="less"`), or $p_i = 2(1 - \Phi(|z_i|))$ (if `alternative="two.sided"`) the corresponding (one- or two-sided) p-value, where $\Phi()$ denotes the cumulative distribution function of a standard normal distribution. Finally, let $w(p_i)$ denote some function that specifies the relative likelihood of selection given the p-value of a study.

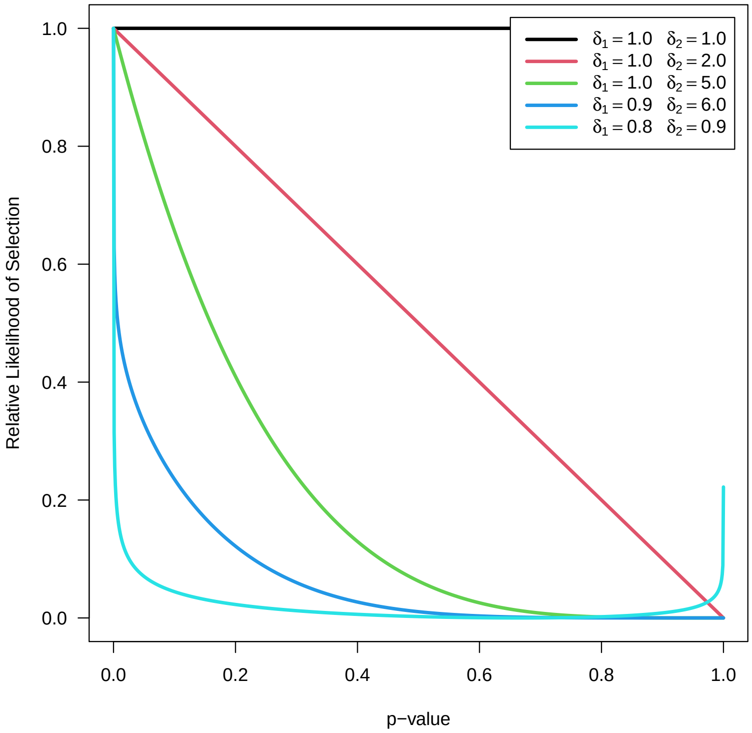
If $w(p_i) > w(p_{i'})$ when $p_i < p_{i'}$ (i.e., $w(p_i)$ is larger for smaller p-values), then `alternative="greater"` implies selection in favor of increasingly significant positive outcomes, `alternative="less"` implies selection in favor of increasingly significant negative outcomes, and `alternative="two.sided"` implies selection in favor of increasingly significant outcomes regardless of their direction.

Beta Selection Model:

When `type="beta"`, the function can be used to fit the ‘beta selection model’ by Citkowicz and Vevea (2017). For this model, the selection function is given by

$$w(p_i) = p_i^{\delta_1 - 1} \times (1 - p_i)^{\delta_2 - 1}$$

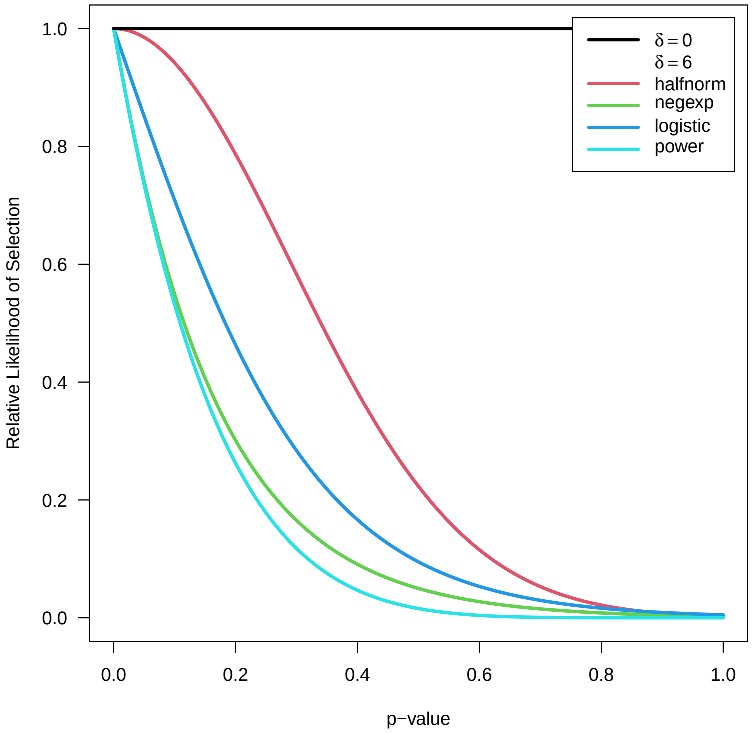
where $\delta_1 > 0$ and $\delta_2 > 0$. The null hypothesis H_0: $\delta_1 = \delta_2 = 1$ represents the case where there is no selection (at least not depending on the p-values). The figure below illustrates with some examples how the relative likelihood of selection can depend on the p-value for various combinations of δ_1 and δ_2. Note that the model allows for a non-monotonic selection function.



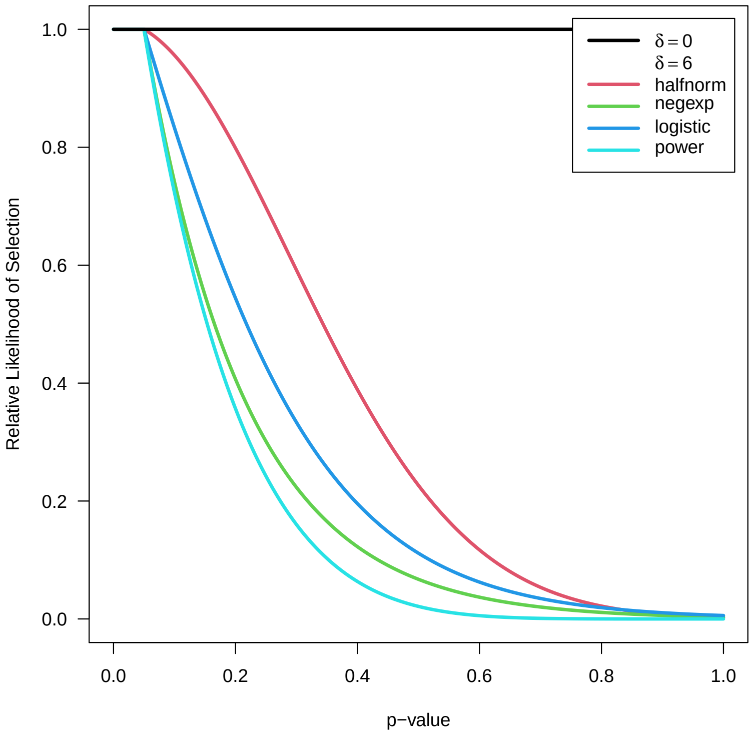
Half-Normal, Negative-Exponential, Logistic, and Power Selection Models:
Preston et al. (2004) suggested the first three of the following selection functions:

name	`type`	**selection function**
half-normal	`"halfnorm"`	$w(p_i) = \exp(-\delta \times p_i^2)$
negative-exponential	`"negexp"`	$w(p_i) = \exp(-\delta \times p_i)$
logistic	`"logistic"`	$w(p_i) = 2 \times \exp(-\delta \times p_i)/(1 + \exp(-\delta \times p_i))$
power	`"power"`	$w(p_i) = (1 - p_i)^\delta$

The power selection model is added here as it has similar properties as the models suggested by Preston et al. (2004). For all models, assume $\delta \geq 0$, so that all functions imply a monotonically decreasing relationship between the p-value and the selection probability. For all functions, H_0: $\delta = 0$ implies no selection. The figure below shows the relative likelihood of selection as a function of the p-value for $\delta = 0$ and for the various selection functions when $\delta = 6$.



Here, these functions are extended to allow for the possibility that $w(p_i) = 1$ for p-values below a certain significance threshold denoted by α (e.g., to model the case that the relative likelihood of selection is equally high for all significant studies but decreases monotonically for p-values above the significance threshold). To fit such a selection model, one should specify the α value (with $0 < \alpha < 1$) via the `steps` argument. There must be at least one observed p-value below and above the chosen threshold to fit these models. The figure below shows some examples of the relative likelihood of selection when `steps=.05`.



Preston et al. (2004) also suggested selection functions where the relatively likelihood of selection not only depends on the p-value, but also the precision (e.g., standard error) of the estimate (if two studies have similar p-values, it may be plausible to assume that the larger / more precise study has a higher probability of selection). These selection functions plus the corresponding power functions are given by:

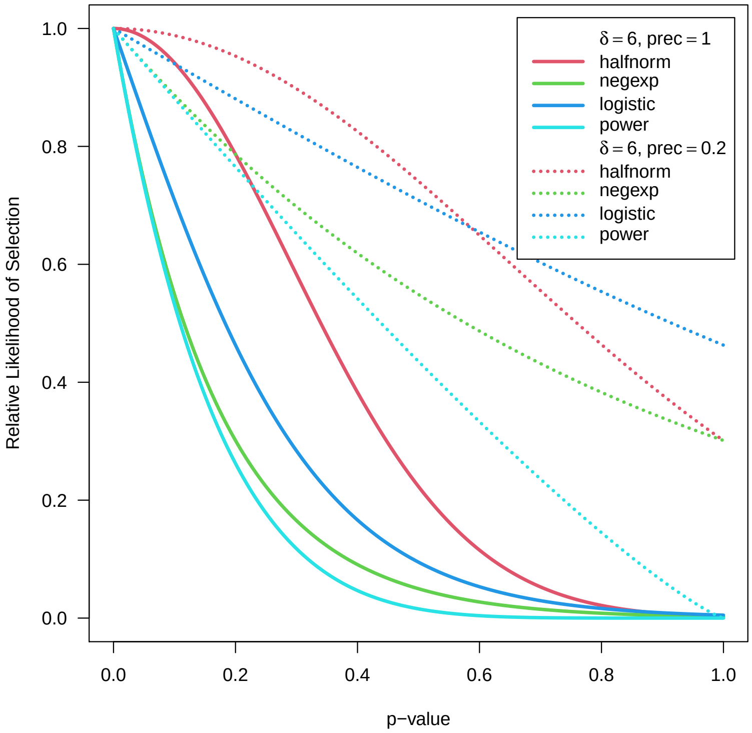
name	`type`	**selection function**
half-normal	`"halfnorm"`	$w(p_i) = \exp(-\delta \times \text{prec}_i \times p_i^2)$
negative-exponential	`"negexp"`	$w(p_i) = \exp(-\delta \times \text{prec}_i \times p_i)$
logistic	`"logistic"`	$w(p_i) = 2 \times \exp(-\delta \times \text{prec}_i \times p_i)/(1 + \exp(-\delta \times \text{prec}_i \times p_i))$
power	`"power"`	$w(p_i) = (1 - p_i)^{-\delta \times \text{prec}_i}$

where $\text{prec}_i = \sqrt{v_i}$ (i.e., the standard error of the ith study) according to Preston et al. (2004). Here, this idea is generalized to allow the user to specify the specific measure of precision to use (via the `prec` argument). Possible options are:

- `prec="sei"` for the standard errors,
- `prec="vi"` for the sampling variances,
- `prec="ninv"` for the inverse of the sample sizes,
- `prec="sqrtninv"` for the inverse square root of the sample sizes.

Using some function of the sample sizes as a measure of precision is only possible when information about the sample sizes is actually stored within the object passed to the `selmodel` function. See ‘Note’.

Note that prec_i is really a measure of imprecision (with higher values corresponding to lower precision). Also, regardless of the specific measure chosen, the values are actually rescaled with $\mathrm{prec}_i = \mathrm{prec}_i / \max(\mathrm{prec}_i)$ inside of the function, such that $\mathrm{prec}_i = 1$ for the least precise study and $\mathrm{prec}_i < 1$ for the remaining studies (the rescaling does not actually change the fit of the model, it only helps to improve the stability of model fitting algorithm). The figure below shows some examples of the relative likelihood of selection using these selection functions for two different precision values.



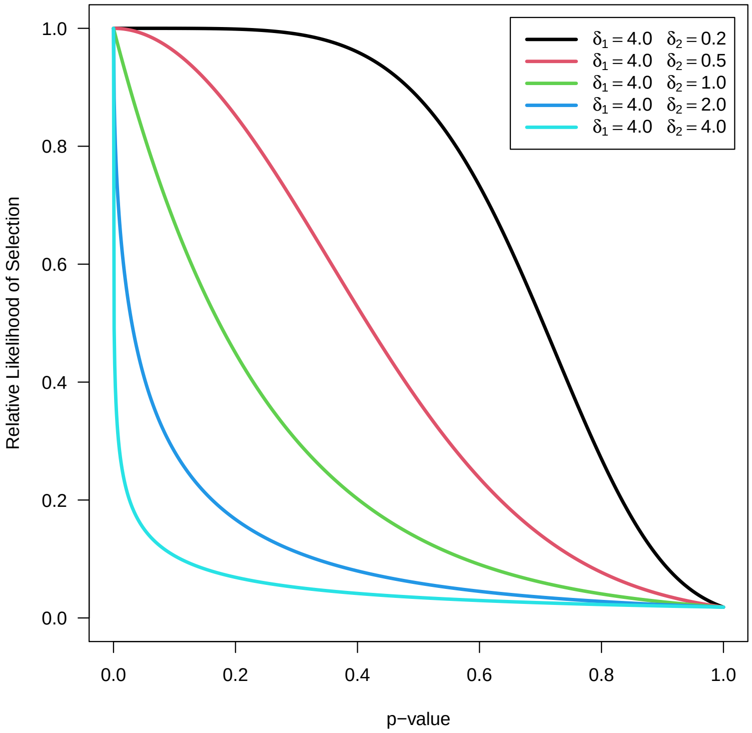
One can also use the `steps` argument as described above in combination with these selection functions (studies with p-values below the chosen threshold then have $w(p_i) = 1$ regardless of their exact p-value or precision).

Negative Exponential Power Selection Model:

As an extension of the half-normal and negative-exponential models, one can also choose `type="negexppow"` for a 'negative exponential power selection model'. The selection function is then given by

$$w(p_i) = \exp(-\delta_1 \times p_i^{1/\delta_2})$$

where $\delta_1 \geq 0$ and $\delta_2 \geq 0$ (see Begg & Mazumdar, 1994, although here a different parameterization is used, such that increasing δ_2 leads to more severe selection). The figure below shows some examples of this selection function when holding δ_1 constant while increasing δ_2.



This model affords greater flexibility in the shape of the selection function, but requires the estimation of the additional power parameter (the half-normal and negative-exponential models are therefore special cases when fixing δ_2 to 0.5 or 1, respectively). H_0: $\delta_1 = 0$ again implies no selection, but so does H_0: $\delta_2 = 0$.

One can again use the `steps` argument to specify a single significance threshold, α, so that $w(p_i) = 1$ for p-values below this threshold and otherwise $w(p_i)$ follows the selection function as given above. One can also use the `prec` argument to specify a measure of precision in combination with this model, which leads to the selection function

$$w(p_i) = \exp(-\delta_1 \times \mathrm{prec}_i \times p_i^{1/\delta_2})$$

and hence is the logical extension of the negative exponential power selection model that also incorporates some measure of precision into the selection process.

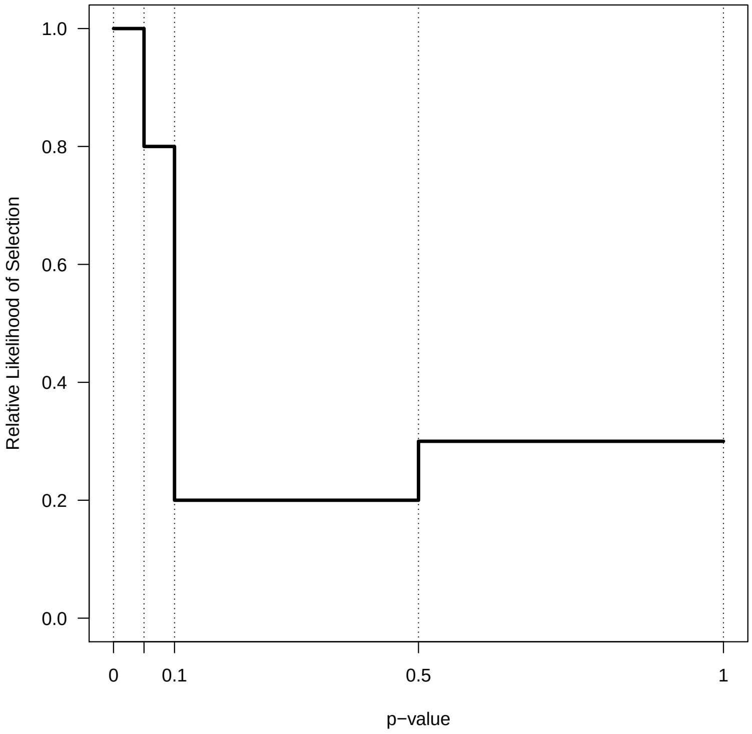
Step Function Selection Models:

When `type="stepfun"`, the function can be used to fit 'step function models' as described by Iyengar and Greenhouse (1988), Hedges (1992), Vevea and Hedges (1995), and Vevea and Woods (2005). For these models, one must specify one or multiple values via the `steps` argument, which define intervals in which the relative likelihood of selection is constant. Let

$$\alpha_1 < \alpha_2 < \ldots < \alpha_c$$

denote these cutpoints sorted in increasing order, with the constraint that $\alpha_c = 1$ (if the highest value specified via `steps` is not 1, the function will automatically add this cutpoint), and define $\alpha_0 = 0$. The selection function is then given by $w(p_i) = \delta_j$ if $\alpha_{j-1} < p_i \leq \alpha_j$. To make the model identifiable, we set $\delta_1 = 1$. The δ_j values therefore denote the likelihood of selection in the various intervals relative to the interval for p-values between 0 and α_1. Hence, the null hypothesis H_0: $\delta_j = 1$ for $j = 1, \ldots, c$ implies no selection.

For example, if `steps=c(.05,.10,.50,1)`, then δ_2 is the likelihood of selection for p-values between .05 and .10, δ_3 is the likelihood of selection for p-values between .10 and .50, and δ_4 is the likelihood of selection for p-values between .50 and 1 relative to the likelihood of selection for p-values between 0 and .05. The figure below shows the corresponding selection function for some arbitrarily chosen δ_j values.



There must be at least one observed p-value within each interval to fit this model. If this is not the case, an error will be issued (setting `verbose=TRUE` provides information about the number of p-values falling into each interval).

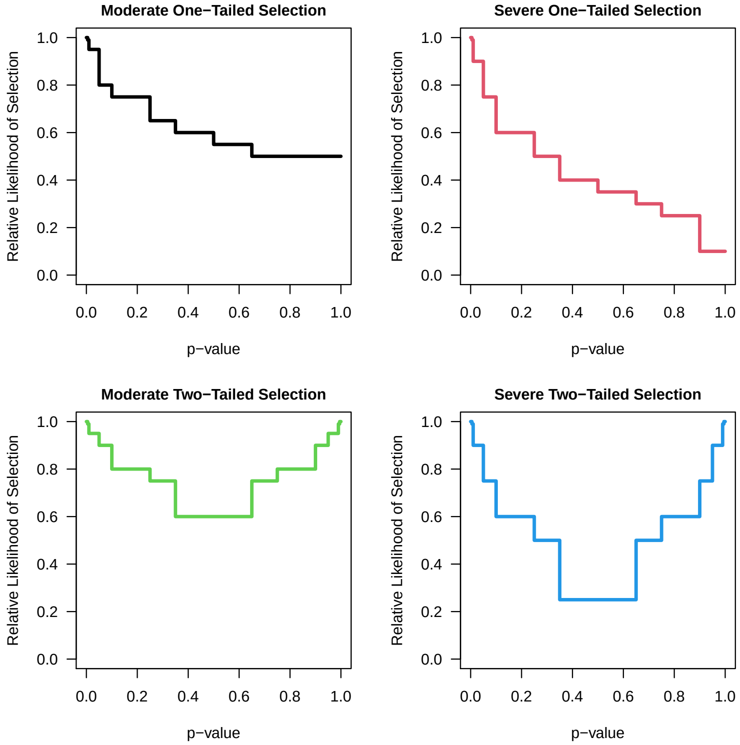
When specifying a single cutpoint in the context of a random-effects model, this model is sometimes called the ‘three-parameter selection model’ (3PSM), corresponding to the parameters μ, τ^2, and δ_2 (e.g., Carter et al., 2019; McShane et al., 2016; Pustejovsky & Rodgers, 2019). The same idea but in the context of a fixed-effects model was also described by Iyengar and Greenhouse (1988).

Note that when `alternative="greater"` or `alternative="less"` (i.e., when we assume that the relative likelihood of selection is not only related to the p-values of the studies, but also the directionality of the outcomes), then it would usually make sense to divide conventional levels of significance (e.g., .05) by 2 before passing these values to the `steps` argument. For example, if we think that studies were selected for positive outcomes that are significant at two-tailed $\alpha = .05$, then we should use `alternative="greater"` in combination with `steps=c(.025,1)`.

One of the challenges when fitting this model with many cutpoints is the large number of parameters that need to be estimated (which is especially problematic when the number of studies is small). An alternative approach suggested by Vevea and Woods (2005) is to fix the δ_j values to some a priori chosen values instead of estimating them. One can then conduct a sensitivity analysis by examining the results (e.g., the estimates of μ and τ^2 in a random-effects model) for a variety of different sets of δ_j values (reflecting more or less severe forms of selection). This can be done by specifying the δ_j values via the `delta` argument. Table 1 in Vevea and Woods (2005) provides some illustrative examples of moderate and severe selection functions for one- and two-tailed selection. The code below creates a data frame that contains these functions.

```
tab <- data.frame(
  steps = c(0.005, 0.01, 0.05, 0.10, 0.25, 0.35, 0.50, 0.65, 0.75, 0.90, 0.95, 0.99, 0.995, 1),
  delta.mod.1 = c(1, 0.99, 0.95, 0.80, 0.75, 0.65, 0.60, 0.55, 0.50, 0.50, 0.50, 0.50, 0.50, 0.50),
  delta.sev.1 = c(1, 0.99, 0.90, 0.75, 0.60, 0.50, 0.40, 0.35, 0.30, 0.25, 0.10, 0.10, 0.10, 0.10),
  delta.mod.2 = c(1, 0.99, 0.95, 0.90, 0.80, 0.75, 0.60, 0.60, 0.75, 0.80, 0.90, 0.95, 0.99, 1.00),
  delta.sev.2 = c(1, 0.99, 0.90, 0.75, 0.60, 0.50, 0.25, 0.25, 0.50, 0.60, 0.75, 0.90, 0.99, 1.00))
```

The figure below shows the corresponding selection functions.



These four functions are “merely examples and should not be regarded as canonical” (Vevea & Woods, 2005).

Value

An object of class `c("rma.uni","rma")`. The object is a list containing the same components as a regular `c("rma.uni","rma")` object, but the parameter estimates are based on the selection model. Most importantly, the following elements are modified based on the selection model:

`beta` estimated coefficients of the model.

`se` standard errors of the coefficients.

`zval` test statistics of the coefficients.

`pval` corresponding p-values.

`ci.lb` lower bound of the confidence intervals for the coefficients.

`ci.ub` upper bound of the confidence intervals for the coefficients.

`vb` variance-covariance matrix of the estimated coefficients.

`tau2` estimated amount of (residual) heterogeneity. Always `0` when `method="FE"`.

`se.tau2` standard error of the estimated amount of (residual) heterogeneity.

In addition, the object contains the following additional elements:

`delta` estimated selection model parameter(s).

`se.delta` corresponding standard error(s).

`zval.delta` corresponding test statistic(s).

`pval.delta` corresponding p-value(s).

`ci.lb.delta` lower bound of the confidence intervals for the parameter(s).

`ci.ub.delta` upper bound of the confidence intervals for the parameter(s).

`LRT` test statistic of the likelihood ratio test for the selection model parameter(s).

`LRTdf` degrees of freedom for the likelihood ratio test.

`LRTp` p-value for the likelihood ratio test.

`LRT.tau2` test statistic of the likelihood ratio test for testing H_0: $\tau^2 = 0$ (`NA` when fitting a fixed-effects model).

`LRTp.tau2` p-value for the likelihood ratio test.

`...` some additional elements/values.

Methods

The results of the fitted model are formatted and printed with the `print.rma.uni` function. The estimated selection function can be drawn with `plot.rma.uni.selmodel`.

The `profile.rma.uni.selmodel` function can be used to obtain a plot of the log-likelihood as a function of τ^2 and/or the selection model parameter(s) of the model. Corresponding confidence intervals can be obtained with the `confint.rma.uni.selmodel` function.

Note

Model fitting is done via numerical optimization over the model parameters. By default, `optim` is used for the optimization. One can also chose a different optimizer via the `control` argument (e.g., `control=list(optimizer="nlminb")`). When using `optim`, one can set the particular method via the `optmethod` argument (e.g., `control=list(optimizer="optim",optmethod="BFGS")`, which is the default). Besides `optim` and `nlminb`, one can also choose one of the optimizers from the `minqa` package (i.e., `uobyqa`, `newuoa`, or `bobyqa`), one of the (derivative-free) algorithms from the `nloptr` package, the Newton-type algorithm implemented in `nlm`, the various algorithms implemented in the `dfoptim` package (`hjk` for the Hooke-Jeeves, `nmk` for the Nelder-Mead, and `mads` for the Mesh Adaptive Direct Searches (MADS) algorithm), the quasi-Newton type optimizer `ucminf` from the package of the same name, or the parallelized version of the L-BFGS-B algorithm implemented in `optimParallel` from the package of the same name.

The optimizer name must be given as a character string (i.e., in quotes). Additional control parameters can be specified via the `control` argument (e.g., `control=list(maxit=1000,reltol=1e-8)`). For `nloptr`, the default is to use the BOBYQA implementation from that package with a relative convergence criterion of `1e-8` on the function value (i.e., log-likelihood), but this can be changed via the `algorithm` and `ftop_rel` arguments (e.g., `control=list(optimizer="nloptr",algorithm="NLOPT_LN_SBPLX",fto` For `optimParallel`, the control argument `ncpus` can be used to specify the number of cores to use for the parallelization (e.g., `control=list(optimizer="optimParallel",ncpus=2)`). With `parallel::detectCores()`, one can check on the number of available cores on the local machine.

All selection models (except for `type="stepfun"`) require repeated evaluations of an integral, which is done via adaptive quadrature as implemented in the `integrate` function. One can adjust the arguments of the `integrate` function via control element `intCtrl`, which is a list of named arguments (e.g., `control = list(intCtrl = list(rel.tol=1e-4,subdivisions=100))`).

The starting values for the fixed effects, the τ^2 value (only relevant in random/mixed-effects models), and the δ parameter(s) are chosen automatically by the function, but one can also set the starting values manually via the `control` argument by specifying a vector of the appropriate length for `beta.init`, a single value for `tau2.init`, and a vector of the appropriate length for `delta.init`.

By default, the δ parameter(s) are constrained to a certain range, which improves the stability of the optimization algorithm. For all models, the maximum is set to `100` and the minimum to `0` (except for `type="beta"`, where the minimum for both parameters is `1e-05`). These defaults can be changed via the `control` argument by specifying a vector of the appropriate length for `delta.min` and/or `delta.max`.

A difficulty with fitting the beta selection model (i.e., `type="beta"`) is the behavior of $w(p_i)$ when $p_i = 0$ or $p_i = 1$. When $\delta_1 < 1$ or $\delta_2 < 1$, then this leads to selection weights equal to infinity, which causes problems when computing the likelihood function. Following Citkowicz and Vevea (2017), this problem can be avoided by censoring p-values too close to 0 or 1. The specific censoring point can be set via the `pval.min` element of the `control` argument. The default for this selection model is `control=list(pval.min=1e-5)`. A similar issues arises for the power selection model (i.e., `type="power"`) when $p_i = 1$. Again, `pval.min=1e-5` is used to circumvent this issue. For all other selection models, the default is `pval.min=0`.

The variance-covariance matrix corresponding to the estimates of the fixed effects, the τ^2 value (only relevant in random/mixed-effects models), and the δ parameter(s) is obtained by inverting the Hessian, which is numerically approximated using the `hessian` function. This may fail, leading to `NA` values for the standard errors and hence test statistics, p-values, and confidence interval bounds. One can set control argument `hessianCtrl` to a list of named arguments to be passed on to the `method.args` argument of the `hessian` function (the default is `control=list(hessianCtrl=list(r=6))`).

Information on the progress of the optimization algorithm can be obtained by setting `verbose=TRUE` (this won't work when using parallelization). This option is useful to determine how the model fitting is progressing. One can also set `verbose` to an integer (`verbose=2` yields even more information and `verbose=3` also show the progress visually by drawing the selection function as the optimization proceeds).

For selection functions where the `prec` argument is relevant, using (a function of) the sample sizes as the measure of precision (i.e., `prec="ninv"` or `prec="sqrtninv"`) is only possible when information about the sample sizes is actually stored within the object passed to the `selmodel` function. That should automatically be the case when the observed effect sizes or outcomes were computed with the `escalc` function or when the observed effect sizes or outcomes were computed within the model fitting function. On the other hand, this will not be the case when `rma.uni` was used together with the `yi` and `vi` arguments and the `yi` and `vi` values were *not* computed with `escalc`. In that case, it is still possible to pass information about the sample sizes to the `rma.uni` function (e.g., use `rma.uni(yi,vi,ni=ni,data=dat)`, where data frame `dat` includes a variable called `ni` with the sample sizes).

Finally, the automatic rescaling of the chosen precision measure can be switched off by setting `scaleprec=FALSE`.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Begg, C. B., & Mazumdar, M. (1994). Operating characteristics of a rank correlation test for publication bias. *Biometrics*, **50**(4), 1088–1101. https://doi.org/10.2307/2533446

Carter, E. C., Schönbrodt, F. D., Gervais, W. M., & Hilgard, J. (2019). Correcting for bias in psychology: A comparison of meta-analytic methods. *Advances in Methods and Practices in Psychological Science*, **2**(2), 115–144. https://doi.org/10.1177/2515245919847196

Citkowicz, M., & Vevea, J. L. (2017). A parsimonious weight function for modeling publication bias. *Psychological Methods*, **22**(1), 28–41. https://doi.org/10.1037/met0000119

Hedges, L. V. (1992). Modeling publication selection effects in meta-analysis. *Statistical Science*, **7**(2), 246–255. https://doi.org/10.1214/ss/1177011364

Iyengar, S., & Greenhouse, J. B. (1988). Selection models and the file drawer problem. *Statistical Science*, **3**(1), 109–117. https://doi.org/10.1214/ss/1177013012

McShane, B. B., Bockenholt, U., & Hansen, K. T. (2016). Adjusting for publication bias in meta-analysis: An evaluation of selection methods and some cautionary notes. *Perspectives on Psychological Science*, **11**(5), 730–749. https://doi.org/10.1177/1745691616662243

Preston, C., Ashby, D., & Smyth, R. (2004). Adjusting for publication bias: Modelling the selection process. *Journal of Evaluation in Clinical Practice*, **10**(2), 313–322. https://doi.org/10.1111/j.1365-2753.2003.00457.x

Pustejovsky, J. E., & Rodgers, M. A. (2019). Testing for funnel plot asymmetry of standardized mean differences. *Research Synthesis Methods*, **10**(1), 57–71. https://doi.org/10.1002/jrsm.1332

Vevea, J. L., & Hedges, L. V. (1995). A general linear model for estimating effect size in the presence of publication bias. *Psychometrika*, **60**(3), 419–435. https://doi.org/10.1007/BF02294384

Vevea, J. L., & Woods, C. M. (2005). Publication bias in research synthesis: Sensitivity analysis using a priori weight functions. *Psychological Methods*, **10**(4), 428–443. https://doi.org/10.1037/1082-989X.10.4.428

See Also

`rma.uni`

Examples

```
############################################################################

### example from Citkowicz and Vevea (2017) for beta selection model

# copy data into 'dat' and examine data
dat <- dat.baskerville2012
dat

# fit random-effects model
res <- rma(smd, se^2, data=dat, method=”ML”, digits=3)
res

# funnel plot
funnel(res, ylim=c(0,0.6), xlab=”Standardized Mean Difference”)

# fit beta selection model
## Not run:
sel <- selmodel(res, type=”beta”)
sel

# plot selection function
plot(sel, ylim=c(0,40))
## End(Not run)

# fit mixed-effects meta-regression model with 'blind' dummy variable as moderator
res <- rma(smd, se^2, data=dat, mods = ~ blind, method=”ML”, digits=3)
res

# predicted average effect for studies that do not and that do use blinding
predict(res, newmods=c(0,1))

# fit beta selection model
## Not run:
sel <- selmodel(res, type=”beta”)
sel
predict(sel, newmods=c(0,1))
## End(Not run)

############################################################################

### example from Preston et al. (2004)

# copy data into 'dat' and examine data
dat <- dat.hahn2001
dat

### meta-analysis of (log) odds rations using the Mantel-Haenszel method
res <- rma.mh(measure=”OR”, ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, digits=2, slab=study)
res

# calculate log odds ratios and corresponding sampling variances
```

```
dat <- escalc(measure="OR", ai=ai, n1i=n1i, ci=ci, n2i=n2i, data=dat, drop00=TRUE)
dat

# fit fixed-effects model
res <- rma(yi, vi, data=dat, method="FE")

# predicted odds ratio (with 95% CI)
predict(res, transf=exp, digits=2)

# funnel plot
funnel(res, atransf=exp, at=log(c(0.01,0.1,1,10,100)), ylim=c(0,2))

# fit half-normal, negative-exponential, logistic, and power selection models
## Not run:
sel1 <- selmodel(res, type="halfnorm", alternative="less")
sel2 <- selmodel(res, type="negexp",   alternative="less")
sel3 <- selmodel(res, type="logistic", alternative="less")
sel4 <- selmodel(res, type="power",    alternative="less")

# plot selection functions
plot(sel1)
plot(sel2, add=TRUE, col="blue")
plot(sel3, add=TRUE, col="red")
plot(sel4, add=TRUE, col="green")

# show estimates of delta (and corresponding SEs)
tab <- data.frame(delta = c(sel1$delta, sel2$delta, sel3$delta, sel4$delta),
                  se    = c(sel1$se.delta, sel2$se.delta, sel3$se.delta, sel4$se.delta))
rownames(tab) <- c("Half-normal", "Negative-exponential", "Logistic", "Power")
round(tab, 2)

# predicted odds ratios (with 95% CI)
predict(res,  transf=exp, digits=2)
predict(sel1, transf=exp, digits=2)
predict(sel2, transf=exp, digits=2)
predict(sel3, transf=exp, digits=2)
predict(sel4, transf=exp, digits=2)
## End(Not run)

# fit selection models including standard error as precision measure (note: using
# scaleprec=FALSE here since Preston et al. (2004) did not use the rescaling)
## Not run:
sel1 <- selmodel(res, type="halfnorm", prec="sei", alternative="less", scaleprec=FALSE)
sel2 <- selmodel(res, type="negexp",   prec="sei", alternative="less", scaleprec=FALSE)
sel3 <- selmodel(res, type="logistic", prec="sei", alternative="less", scaleprec=FALSE)
sel4 <- selmodel(res, type="power",    prec="sei", alternative="less", scaleprec=FALSE)

# show estimates of delta (and corresponding SEs)
tab <- data.frame(delta = c(sel1$delta, sel2$delta, sel3$delta, sel4$delta),
                  se    = c(sel1$se.delta, sel2$se.delta, sel3$se.delta, sel4$se.delta))
rownames(tab) <- c("Half-normal", "Negative-exponential", "Logistic", "Power")
round(tab, 2)
```

```
# predicted odds ratio (with 95% CI)
predict(res,  transf=exp, digits=2)
predict(sel1, transf=exp, digits=2)
predict(sel2, transf=exp, digits=2)
predict(sel3, transf=exp, digits=2)
predict(sel4, transf=exp, digits=2)
## End(Not run)

############################################################################

### meta-analysis on the effect of environmental tobacco smoke on lung cancer risk

# copy data into 'dat' and examine data
dat <- dat.hackshaw1998
dat

# fit random-effects model
res <- rma(yi, vi, data=dat, method=”ML”)
res

# funnel plot
funnel(res, atransf=exp, at=log(c(0.25,0.5,1,2,4,8)), ylim=c(0,0.8))

# step function selection model
## Not run:
sel <- selmodel(res, type=”stepfun”, alternative=”greater”, steps=c(.025,.10,.50,1))
sel

# plot selection function
plot(sel)
## End(Not run)

############################################################################

### validity of student ratings example from Vevea & Woods (2005)

# copy data into 'dat' and examine data
dat <- dat.cohen1981
dat

# calculate r-to-z transformed correlations and corresponding sampling variances
dat <- escalc(measure=”ZCOR”, ri=ri, ni=ni, data=dat[c(1,4,5)])
dat

# fit random-effects model
res <- rma(yi, vi, data=dat, method=”ML”, digits=3)
res

# predicted average correlation (with 95% CI)
predict(res, transf=transf.ztor)

# funnel plot
funnel(res, ylim=c(0,0.4))
```

```
# selection functions from Vevea & Woods (2005)
tab <- data.frame(
  steps = c(0.005, 0.01, 0.05, 0.10, 0.25, 0.35, 0.50, 0.65, 0.75, 0.90, 0.95, 0.99, 0.995, 1),
  delta.mod.1 = c(1, 0.99, 0.95, 0.80, 0.75, 0.65, 0.60, 0.55, 0.50, 0.50, 0.50, 0.50, 0.50, 0.50),
  delta.sev.1 = c(1, 0.99, 0.90, 0.75, 0.60, 0.50, 0.40, 0.35, 0.30, 0.25, 0.10, 0.10, 0.10, 0.10),
  delta.mod.2 = c(1, 0.99, 0.95, 0.90, 0.80, 0.75, 0.60, 0.60, 0.75, 0.80, 0.90, 0.95, 0.99, 1.00),
  delta.sev.2 = c(1, 0.99, 0.90, 0.75, 0.60, 0.50, 0.25, 0.25, 0.50, 0.60, 0.75, 0.90, 0.99, 1.00))

# apply step function model with a priori chosen selection weights
## Not run:
sel <- lapply(tab[-1], function(delta) selmodel(res, type="stepfun", steps=tab$steps, delta=delta))

# estimates (transformed correlation) and tau^2 values
sav <- data.frame(estimate = round(c(res$beta, sapply(sel, function(x) x$beta)), 2),
                  varcomp  = round(c(res$tau2, sapply(sel, function(x) x$tau2)), 3))
sav
## End(Not run)

############################################################################
```

---

simulate.rma *Simulate Method for 'rma' Objects*

Description

The function simulates effect sizes or outcomes based on "rma" model object.

Usage

```
## S3 method for class 'rma'
simulate(object, nsim = 1, seed = NULL, olim, ...)
```

Arguments

object	an object of class "rma".
nsim	number of response vectors to simulate (defaults to 1).
seed	an object to specify if and how the random number generator should be initialized ('seeded'). Either NULL or an integer that will be used in a call to set.seed before simulating the response vectors. If set, the value is saved as the "seed" attribute of the returned value. The default, NULL will not change the random generator state, and return .Random.seed as the "seed" attribute; see 'Value'.
olim	optional argument to specify observation/outcome limits for the simulated values. If unspecified, no limits are used.
...	other arguments.

Details

The model specified via object must be a model fitted with either the rma.uni or rma.mv function.

Value

A data frame with nsim columns with the simulated effect sizes or outcomes.

The data frame comes with an attribute "seed". If argument seed is NULL, the attribute is the value of .Random.seed before the simulation was started; otherwise it is the value of the seed argument with a "kind" attribute with value as.list(RNGkind()).

Note

If the outcome measure used for the analysis is bounded (e.g., correlations are bounded between -1 and +1, proportions are bounded between 0 and 1), one can use the olim argument to enforce those observation/outcome limits when simulating values (simulated values cannot exceed those bounds then).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mv

Examples

```
### copy BCG vaccine data into 'dat'
dat <- dat.bcg

### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat)
dat

### fit random-effects model
res <- rma(yi, vi, data=dat)
res

### simulate 10 sets of new outcomes based on the fitted model
newdat <- simulate(res, nsim=10, seed=1234)
newdat
```

tes *Test of Excess Significance*

Description

Function to conduct the test of excess significance.

Usage

```
tes(x, ...)

## S3 method for class 'rma'
tes(x, H0=0, alternative="two.sided", alpha=.05,
    test, tes.alternative="greater", progbar=TRUE, tes.alpha=.10, digits, ...)

## Default S3 method:
tes(x, vi, sei, subset, H0=0, alternative="two.sided", alpha=.05, theta, tau2,
    test, tes.alternative="greater", progbar=TRUE, tes.alpha=.10, digits, ...)

## S3 method for class 'tes'
print(x, digits=x$digits, ...)
```

Arguments

These arguments pertain to data input:

an object of class "rma" or a vector with the observed effect sizes or outcomes.

`xi` vector with the corresponding sampling variances.

`sei` vector with the corresponding standard errors (note: only one of the two, `vi` or `sei`, needs to be specified).

`subset` optional (logical or numeric) vector to specify the subset of studies that should be included.

These arguments pertain to the tests of the observed effect sizes or outcomes:

`H0` numeric value to specify the value of the effect size or outcome under the null hypothesis (the default is 0).

`alternative` character string to specify the sidedness of the hypothesis when testing the observed effect sizes or outcomes. Possible options are "two.sided" (the default), "greater", or "less". Can be abbreviated.

`alpha` alpha level for testing the observed effect sizes or outcomes (the default is .05).

These arguments pertain to the power of the tests:

`theta` numeric value to specify the value of the true effect size or outcome under the alternative hypothesis. If unspecified, it will be estimated based on the data or the value is taken from the "rma" object.

`tau2` numeric value to specify the amount of heterogeneity in the true effect sizes or outcomes. If unspecified, the true effect sizes or outcomes are assumed to be homogeneous or the value is taken from the `"rma"` object.

These arguments pertain to the test of excess significance:

`test` character string to specify the type of test to use for conducting the test of excess significance. Possible options are `"chi2"`, `"binom"`, or `"exact"`. Can be abbreviated. If unspecified, the function chooses the type of test based on the data.

`tes.alternative` character string to specify the sidedness of the hypothesis for the test of excess significance. Possible options are `"greater"` (the default), `"two.sided"`, or `"less"`. Can be abbreviated.

`progbar` logical to specify whether a progress bar should be shown (the default is `TRUE`). Only relevant when conducting an exact test.

`tes.alpha` alpha level for the test of excess significance (the default is .10). Only relevant for finding the ‘limit estimate’.

Miscellaneous arguments:

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is 4.

`...` other arguments.

Details

The function carries out the test of excess significance described by Ioannidis and Trikalinos (2007). The test can be used to examine whether the observed number of significant findings is greater than the number of significant findings expected given the power of the tests. An overabundance of significant tests may suggest that the collection of studies is not representative of all studies conducted on a particular topic.

One can either pass an object of class `"rma"` to the function or a vector with the observed effect sizes or outcomes (via `x`) and the corresponding sampling variances via `vi` (or the standard errors via `sei`).

The observed effect sizes or outcomes are tested for significance based on a standard Wald-type test, that is, by comparing

$$z_i = \frac{y_i - \mathrm{H}_0}{\sqrt{v_i}}$$

against the appropriate critical value(s) of a standard normal distribution (e.g., ± 1.96 for `alternative="two.sided"` and `alpha=.05`, which are the defaults). Let O denote the observed number of significant tests.

Given a particular value for the true effect or outcome denoted by θ (which, if it is unspecified, is determined by computing the inverse-variance weighted average of the observed effect sizes or outcomes or the value is taken from the model object), let $1 - \beta_i$ denote the power of the ith test (where β_i denotes the Type II error probability). If $\tau^2 > 0$, let $1 - \beta_i$ denote the expected power (computed based on integrating the power over a normal distribution with mean θ and variance τ^2). Let $E = \sum_{i=1}^{k}(1 - \beta_i)$ denote the expected number of significant tests.

The test of excess significance then tests if O is significantly greater (if `tes.alternative="greater"`) than E. This can be done using Pearson’s chi-squared test (if `test="chi2"`), a binomial test (if `test="binomial"`), or an exact test (if `test="exact"`). The latter is described in Francis (2013). If argument `test` is unspecified, the default is to do an exact test if the number of elements in the sum that needs to be computed is less than or equal to `10^6` and to do a chi-square test otherwise.

One can also iteratively find the value of θ such that the p-value of the test of excess significance is equal to `tes.alpha` (which is `.10` by default). The resulting value is called the 'limit estimate' and is denoted θ_{lim} by Ioannidis and Trikalinos (2007). Note that the limit estimate is not computable if the p-value is larger than `tes.alpha` even if $\theta = \mathrm{H}_0$.

Value

An object of class `"tes"`. The object is a list containing the following components:

`k` the number of studies included in the analysis.

`O` the observed number of significant tests.

`E` the expected number of significant tests.

`OEratio` the ratio of O over E.

`test` the type of test conducted.

`pval` the p-value of the test of excess significance.

`power` the (estimated) power of the tests.

`sig` logical vector indicating which tests were significant.

`theta` the value of θ used for computing the power of the tests.

`theta.lim` the 'limit estimate' (i.e., θ_{lim}).

`...` some additional elements/values.

The results are formatted and printed with the `print.tes` function.

Note

When `tes.alternative="greater"` (the default), then the function tests if O is significantly greater than E and hence this is indeed a test of excess significance. When `tes.alternative="two.sided"`, then the function tests if O differs significantly from E in either direction and hence it would be more apt to describe this as a test of (in)consistency (between O and E). Finally, one can also set `tes.alternative="less"`, in which case the function tests if O is significantly lower than E, which could be considered a test of excess non-significance.

When `tes.alternative="two.sided"`, one can actually compute two limit estimates. The function attempts to compute both.

The function computes the significance and power of the studies based on Wald-type tests regardless of the effect size or outcome measure used as input. This works as an adequate approximation as long as the within-study sample sizes are not too small.

Note that the test is not a test for publication bias but a test whether the set of studies includes an unusual number of significant findings given the power of the studies. The general usefulness of the test and its usefulness under particular circumstances (e.g., when there is substantial heterogeneity in the true effect sizes or outcomes) has been the subject of considerable debate. See Francis (2013) and the commentaries on this article in the same issue of the journal.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Francis, G. (2013). Replication, statistical consistency, and publication bias. *Journal of Mathematical Psychology*, **57**(5), 153–169. https://doi.org/10.1016/j.jmp.2013.02.003

Ioannidis, J. P. A., & Trikalinos, T. A. (2007). An exploratory test for an excess of significant findings. *Clinical Trials*, **4**(3), 245–253. https://doi.org/10.1177/1740774507079441

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

ranktest, regtest, trimfill

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=x.a, n1i=n.a, ci=x.p, n2i=n.p, data=dat.dorn2007)

### conduct test of excess significance (using test="chi2" to speed things up)
tes(dat$yi, dat$vi, test="chi2")

### same as fitting a FE model and then passing the object to the function
res <- rma(yi, vi, data=dat, method="FE")
tes(res, test="chi2")

### illustrate limit estimate (value of theta where p-value of test is equal to tes.alpha)
thetas <- seq(0,1,length=101)
pvals <- sapply(thetas, function(theta) tes(dat$yi, dat$vi, test="chi2", theta=theta)$pval)
plot(thetas, pvals, type="o", pch=19, ylim=c(0,1))
sav <- tes(dat$yi, dat$vi, test="chi2")
abline(h=sav$tes.alpha, lty="dotted")
abline(v=sav$theta.lim, lty="dotted")

### examine significance of test as a function of alpha (to examine 'significance chasing')
alphas <- seq(.01,.99,length=101)
pvals <- sapply(alphas, function(alpha) tes(dat$yi, dat$vi, test="chi2", alpha=alpha)$pval)
plot(alphas, pvals, type="o", pch=19, ylim=c(0,1))
abline(v=.05, lty="dotted")
abline(h=.10, lty="dotted")
```

---

to.long *Convert Data from Vector to Long Format*

Description

The function converts summary data in vector format to the corresponding long format.

Usage

```
to.long(measure, ai, bi, ci, di, n1i, n2i, x1i, x2i, t1i, t2i,
        m1i, m2i, sd1i, sd2i, xi, mi, ri, ti, sdi, ni, data, slab, subset,
        add=1/2, to="none", drop00=FALSE, vlong=FALSE, append=TRUE, var.names)
```

Arguments

measure — a character string to specify the effect size or outcome measure corresponding to the summary data supplied. See 'Details' and the documentation of the escalc function for possible options.

ai — vector to specify the 2×2 table frequencies (upper left cell).

bi — vector to specify the 2×2 table frequencies (upper right cell).

ci — vector to specify the 2×2 table frequencies (lower left cell).

di — vector to specify the 2×2 table frequencies (lower right cell).

n1i — vector to specify the group sizes or row totals (first group/row).

n2i — vector to specify the group sizes or row totals (second group/row).

x1i — vector to specify the number of events (first group).

x2i — vector to specify the number of events (second group).

t1i — vector to specify the total person-times (first group).

t2i — vector to specify the total person-times (second group).

m1i — vector to specify the means (first group or time point).

m2i — vector to specify the means (second group or time point).

sd1i — vector to specify the standard deviations (first group or time point).

sd2i — vector to specify the standard deviations (second group or time point).

xi — vector to specify the frequencies of the event of interest.

mi — vector to specify the frequencies of the complement of the event of interest or the group means.

ri — vector to specify the raw correlation coefficients.

ti — vector to specify the total person-times.

sdi — vector to specify the standard deviations.

ni — vector to specify the sample/group sizes.

data — optional data frame containing the variables given to the arguments above.

slab — optional vector with labels for the studies.

subset — optional (logical or numeric) vector to specify the subset of studies that should included in the data frame returned by the function.

add — see the documentation of the escalc function.

to — see the documentation of the escalc function.

drop00 — see the documentation of the escalc function.

vlong — optional logical whether a very long format should be used (only relevant for 2×2 or 1×2 table data).

append logical to specify whether the data frame specified via the data argument (if one has been specified) should be returned together with the long format data (the default is TRUE).

var.names optional vector with variable names (length depends on the data type). If unspecified, the function sets appropriate variable names by default.

Details

The escalc function describes a wide variety of effect size or outcome measures that can be computed for a meta-analysis. The summary data used to compute those measures are typically contained in vectors, each element corresponding to a study. The to.long function takes this information and constructs a long format dataset from these data.

For example, in various fields (such as the health and medical sciences), the response variable measured is often dichotomous (binary), so that the data from a study comparing two different groups can be expressed in terms of a 2×2 table, such as:

	outcome 1	outcome 2	total
group 1	ai	bi	n1i
group 2	ci	di	n2i

where ai, bi, ci, and di denote the cell frequencies (i.e., the number of people falling into a particular category) and n1i and n2i the row totals (i.e., the group sizes).

The cell frequencies in k such 2×2 tables can be specified via the ai, bi, ci, and di arguments (or alternatively, via the ai, ci, n1i, and n2i arguments). The function then creates the corresponding long format dataset. The measure argument should then be set equal to one of the outcome measures that can be computed based on this type of data, such as "RR", "OR", "RD" (it is not relevant which specific measure is chosen, as long as it corresponds to the specified summary data). See the documentation of the escalc function for more details on the types of data formats available.

The long format for data of this type consists of two rows per study, a factor indicating the study (default name study), a dummy variable indicating the group (default name group, coded as 1 and 2), and two variables indicating the number of individuals experiencing outcome 1 or outcome 2 (default names out1 and out2). Alternatively, if vlong=TRUE, then the long format consists of four rows per study, a factor indicating the study (default name study), a dummy variable indicating the group (default name group, coded as 1 and 2), a dummy variable indicating the outcome (default name outcome, coded as 1 and 2), and a variable indicating the frequency of the respective outcome (default name freq).

The default variable names can be changed via the var.names argument (must be of the appropriate length, depending on the data type).

The examples below illustrate the use of this function.

Value

A data frame with either k, $2 \times k$, or $4 \times k$ rows and an appropriate number of columns (depending on the data type) with the data in long format. If append=TRUE and a data frame was specified via the data argument, then the data in long format are appended to the original data frame (with rows repeated an appropriate number of times).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

escalc, to.table

Examples

```
### convert data to long format
dat <- to.long(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)
dat

### extra long format
dat <- to.long(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, vlong=TRUE)
dat

### convert data to long format
dat <- to.long(measure="IRR", x1i=x1i, x2i=x2i, t1i=t1i, t2i=t2i,
               data=dat.hart1999, var.names=c("id", "group", "events", "ptime"))
dat

### convert data to long format
dat <- to.long(measure="MD", m1i=m1i, sd1i=sd1i, n1i=n1i,
               m2i=m2i, sd2i=sd2i, n2i=n2i, data=dat.normand1999,
               var.names=c("id", "group", "mean", "sd", "n"))
dat
```

---

to.table *Convert Data from Vector to Table Format*

Description

The function converts summary data in vector format to the corresponding table format.

Usage

```
to.table(measure, ai, bi, ci, di, n1i, n2i, x1i, x2i, t1i, t2i,
         m1i, m2i, sd1i, sd2i, xi, mi, ri, ti, sdi, ni, data, slab, subset,
         add=1/2, to="none", drop00=FALSE, rows, cols)
```

Arguments

`measure` a character string to specify the effect size or outcome measure corresponding to the summary data supplied. See ‘Details’ and the documentation of the `escalc` function for possible options.

`ai` vector to specify the 2×2 table frequencies (upper left cell).

`bi` vector to specify the 2×2 table frequencies (upper right cell).

`ci` vector to specify the 2×2 table frequencies (lower left cell).

`di` vector to specify the 2×2 table frequencies (lower right cell).

`n1i` vector to specify the group sizes or row totals (first group/row).

`n2i` vector to specify the group sizes or row totals (second group/row).

`x1i` vector to specify the number of events (first group).

`x2i` vector to specify the number of events (second group).

`t1i` vector to specify the total person-times (first group).

`t2i` vector to specify the total person-times (second group).

`m1i` vector to specify the means (first group or time point).

`m2i` vector to specify the means (second group or time point).

`sd1i` vector to specify the standard deviations (first group or time point).

`sd2i` vector to specify the standard deviations (second group or time point).

`xi` vector to specify the frequencies of the event of interest.

`mi` vector to specify the frequencies of the complement of the event of interest or the group means.

`ri` vector to specify the raw correlation coefficients.

`ti` vector to specify the total person-times.

`sdi` vector to specify the standard deviations.

`ni` vector to specify the sample/group sizes.

`data` optional data frame containing the variables given to the arguments above.

`slab` optional vector with labels for the studies.

`subset` optional (logical or numeric) vector to specify the subset of studies that should be included in the array returned by the function.

`add` see the documentation of the `escalc` function.

`to` see the documentation of the `escalc` function.

`drop00` see the documentation of the `escalc` function.

`rows` optional vector with row/group names.

`cols` optional vector with column/outcome names.

Details

The escalc function describes a wide variety of effect size or outcome measures that can be computed for a meta-analysis. The summary data used to compute those measures are typically contained in vectors, each element corresponding to a study. The to.table function takes this information and constructs an array of k tables from these data.

For example, in various fields (such as the health and medical sciences), the response variable measured is often dichotomous (binary), so that the data from a study comparing two different groups can be expressed in terms of a 2×2 table, such as:

	outcome 1	outcome 2	total
group 1	ai	bi	n1i
group 2	ci	di	n2i

where ai, bi, ci, and di denote the cell frequencies (i.e., the number of people falling into a particular category) and n1i and n2i the row totals (i.e., the group sizes).

The cell frequencies in k such 2×2 tables can be specified via the ai, bi, ci, and di arguments (or alternatively, via the ai, ci, n1i, and n2i arguments). The function then creates the corresponding $2 \times 2 \times k$ array of tables. The measure argument should then be set equal to one of the outcome measures that can be computed based on this type of data, such as "RR", "OR", "RD" (it is not relevant which specific measure is chosen, as long as it corresponds to the specified summary data). See the documentation of the escalc function for more details on the types of data formats available.

The examples below illustrate the use of this function.

Value

An array with k elements each consisting of either 1 or 2 rows and an appropriate number of columns.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

escalc, to.long

Examples

```
### create tables
dat <- to.table(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg,
                data=dat.bcg, slab=paste(author, year, sep=", "),
                rows=c("Vaccinated", "Not Vaccinated"), cols=c("TB+", "TB-"))
dat
```

```
### create tables
dat <- to.table(measure="IRR", x1i=x1i, x2i=x2i, t1i=t1i, t2i=t2i,
                data=dat.hart1999, slab=paste(study, year, sep=", "),
                rows=c("Warfarin Group", "Placebo/Control Group"))
dat

### create tables
dat <- to.table(measure="MD", m1i=m1i, sd1i=sd1i, n1i=n1i,
                m2i=m2i, sd2i=sd2i, n2i=n2i, data=dat.normand1999,
                slab=source, rows=c("Specialized Care", "Routine Care"))
dat
```

to.wide *Convert Data from a Long to a Wide Format*

Description

The function converts data given in long format to a wide format.

Usage

```
to.wide(data, study, grp, ref, grpvars, postfix=c(".1",".2"),
        addid=TRUE, addcomp=TRUE, adddesign=TRUE, minlen=2,
        var.names=c("id","comp","design"))
```

Arguments

`data` a data frame in long format.

`study` either the name (given as a character string) or the position (given as a single number) of the study variable in the data frame.

`grp` either the name (given as a character string) or the position (given as a single number) of the group variable in the data frame.

`ref` optional character string to specify the reference group (must be one of the groups in the group variable). If not given, the most frequently occurring group is used as the reference group.

`grpvars` either the names (given as a character vector) or the positions (given as a numeric vector) of the group-level variables.

`postfix` a character string of length 2 giving the affix that is placed after the names of the group-level variables for the first and second group.

`addid` logical to specify whether a row id variable should be added to the data frame (the default is `TRUE`).

`addcomp` logical to specify whether a comparison id variable should be added to the data frame (the default is `TRUE`).

`adddesign` logical to specify whether a design id variable should be added to the data frame (the default is `TRUE`).

`minlen`	integer to specify the minimum length of the shortened group names for the comparison and design id variables (the default is 2).
`var.names`	a character string with three elements to specify the name of the id, comparison, and design variables (the defaults are `"id"`, `"comp"`, and `"design"`, respectively).

Details

A meta-analytic dataset may be structured in a 'long' format, where each row in the dataset corresponds to a particular study group (e.g., treatment arm). Using this function, such a dataset can be restructured into a 'wide' format, where each group within a study is contrasted against a particular reference group.

The `study` and `group` arguments are used to specify the study and group variables in the dataset (either as character strings or as numbers indicating the column positions of these variables in the dataset). Optional argument `ref` is used to specify the reference group (this must be one of the groups in the `group` variable). Argument `grpvars` is used to specify (either as a character vector or by giving the column positions) of those variables in the dataset that correspond to group-level outcomes (the remaining variables are treated as study-level outcomes).

The dataset is restructured so that a two-group study will yield a single row in the restructured dataset, contrasting the first group against the second/reference group. For studies with more than two groups (often called 'multiarm' or 'multitreatment' studies in the medical literature), the reference group is repeated as many times as needed (so a three-group study would yield two rows in the restructured dataset, contrasting two groups against a common reference group).

If a study does not include the reference group, then another group from the study will be used as the reference group. This group is chosen based on the factor levels of the `grp` variable (i.e., the last level that occurs in the study becomes the reference group).

To distinguish the names of the group-level outcome variables for the two first and second group in the restructured dataset, the strings given for the `postfix` argument are placed after the respective variable names.

If requested, row id, comparison id, and design id variables are added to the restructured dataset. The row id is simply a unique number for each row in the dataset. The comparison id variable indicates which two groups have been compared against each other). The design id variable indicates which groups were included in a particular study. The group names are shortened for the comparison and design variables (to at least `minlen`; the actual length might be longer to ensure uniqueness of the group names).

The examples below illustrate the use of this function.

Value

A data frame with rows contrasting groups against a reference group and an appropriate number of columns (depending on the number of group-level outcome variables).

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

contrmat, dat.hasselblad1998, dat.lopez2019, dat.obrien2003, dat.pagliaro1992, dat.senn2013

Examples

```
### data in long format
dat <- dat.senn2013
dat <- dat[c(1,4,3,2,5,6)]
dat

### restructure to wide format
dat <- to.wide(dat, study="study", grp="treatment", ref="placebo", grpvars=4:6)
dat

### data in long format
dat <- dat.hasselblad1998
dat

### restructure to wide format
dat <- to.wide(dat, study="study", grp="trt", ref="no_contact", grpvars=6:7)
dat
```

transf *Transformation Functions*

Description

A set of transformation functions useful for meta-analyses.

Usage

```
transf.rtoz(xi, ...)
transf.ztor(xi, ...)
transf.logit(xi, ...)
transf.ilogit(xi, ...)
transf.arcsin(xi, ...)
transf.iarcsin(xi, ...)
transf.pft(xi, ni, ...)
transf.ipft(xi, ni, ...)
transf.ipft.hm(xi, targs, ...)
transf.isqrt(xi, ...)
transf.irft(xi, ti, ...)
transf.iirft(xi, ti, ...)
transf.ahw(xi, ...)
transf.iahw(xi, ...)
transf.abt(xi, ...)
transf.iabt(xi, ...)
transf.ztor.int(xi, targs, ...)
transf.exp.int(xi, targs, ...)
transf.ilogit.int(xi, targs, ...)
transf.dtou1(xi, ...)
transf.dtou2(xi, ...)
transf.dtou3(xi, ...)
transf.dtorpb(xi, n1i, n2i, ...)
transf.dtobesd(xi, ...)
transf.dtomd(xi, targs, ...)
transf.logortord(xi, pc, ...)
transf.logortorr(xi, pc, ...)
```

Arguments

`xi`	vector of values to be transformed.
`ni`	vector of sample sizes.
`n1i`	vector of sample sizes for the first group.
`n2i`	vector of sample sizes for the second group.
`ti`	vector of person-times at risk.
`pc`	control group risk (either a single value or a vector).
`targs`	list with additional arguments for the transformation function. See 'Details'.
`...`	other arguments.

Details

The following transformation functions are currently implemented:

- `transf.rtoz`: Fisher's r-to-z transformation for correlations.
- `transf.ztor`: inverse of Fisher's r-to-z transformation.
- `transf.logit`: logit (log odds) transformation for proportions.
- `transf.ilogit`: inverse of the logit transformation.
- `transf.arcsin`: arcsine square root transformation for proportions.
- `transf.iarcsin`: inverse of the arcsine transformation.
- `transf.pft`: Freeman-Tukey (double arcsine) transformation for proportions. See Freeman & Tukey (1950). The `xi` argument is used to specify the proportions and the `ni` argument the corresponding sample sizes.
- `transf.ipft`: inverse of the Freeman-Tukey (double arcsine) transformation for proportions. See Miller (1978).
- `transf.ipft.hm`: inverse of the Freeman-Tukey (double arcsine) transformation for proportions using the harmonic mean of the sample sizes for the back-transformation. See Miller (1978). The sample sizes are specified via the `targs` argument (the list element should be called `ni`).

- `transf.isqrt`: inverse of the square root transformation (i.e., function to square a number).
- `transf.irft`: Freeman-Tukey transformation for incidence rates. See Freeman & Tukey (1950). The `xi` argument is used to specify the incidence rates and the `ti` argument the corresponding person-times at risk.
- `transf.iirft`: inverse of the Freeman-Tukey transformation for incidence rates.
- `transf.ahw`: transformation of coefficient alpha as suggested by Hakstian & Whalen (1976).
- `transf.iahw`: inverse of the transformation of coefficient alpha as suggested by Hakstian & Whalen (1976).
- `transf.abt`: transformation of coefficient alpha as suggested by Bonett (2002).
- `transf.iabt`: inverse of the transformation of coefficient alpha as suggested by Bonett (2002).
- `transf.ztor.int`: integral transformation method for the z-to-r transformation.
- `transf.exp.int`: integral transformation method for the exponential transformation.
- `transf.ilogit.int`: integral transformation method for the inverse of the logit transformation.
- `transf.dtou1`: transformation of standardized mean differences to Cohen's U1 values (Cohen, 1988).
- `transf.dtou2`: transformation of standardized mean differences to Cohen's U2 values (Cohen, 1988).
- `transf.dtou3`: transformation of standardized mean differences to Cohen's U3 values (Cohen, 1988).
- `transf.dtocles`: transformation of standardized mean differences to common language effect size values (McGraw & Wong, 1992).
- `transf.dtorpb`: transformation of standardized mean differences to point-biserial correlations. If `n1i` and `n2i` are not specified, the function assumes `n1i=n2i` and uses an approximate formula. If `n1i` and `n2i` are specified, the exact transformation formula is used.
- `transf.dtobesd`: transformation of standardized mean differences to binomial effect size display values (Rosenthal & Rubin, 1982). Note that the function only provides the proportion in the first group scoring above the median (the proportion in the second group scoring above the median is simply one minus the proportion in the first group scoring above the median).
- `transf.dtomd`: transformation of standardized mean differences to mean differences given a known standard deviation, which must be specified via the `targs` argument.
- `transf.logortord`: transformation of log odds ratios to risk differences, assuming a particular value for the control group risk (which needs to be specified via the `pc` argument).
- `transf.logortorr`: transformation of log odds ratios to risk ratios, assuming a particular value for the control group risk (which needs to be specified via the `pc` argument).

Value

A vector with the transformed values.

Note

The integral transformation method for a transformation function $h(z)$ integrates $h(z)f(z)$ over z using the limits `targs$lower` and `targs$upper`, where $f(z)$ is the density of a normal distribution with mean equal to `xi` and variance equal to `targs$tau2`. An example is provided below.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Bonett, D. G. (2002). Sample size requirements for testing and estimating coefficient alpha. *Journal of Educational and Behavioral Statistics*, **27**(4), 335–340. https://doi.org/10.3102/10769986027004335

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

Fisher, R. A. (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron*, **1**, 1–32. http://hdl.handle.net/2440/15169

Freeman, M. F., & Tukey, J. W. (1950). Transformations related to the angular and the square root. *Annals of Mathematical Statistics*, **21**(4), 607–611. https://doi.org/10.1214/aoms/1177729756

Hakstian, A. R., & Whalen, T. E. (1976). A k-sample significance test for independent alpha coefficients. *Psychometrika*, **41**(2), 219–231. https://doi.org/10.1007/BF02291840

McGraw, K. O., & Wong, S. P. (1992). A common language effect size statistic. *Psychological Bulletin*, **111**(2), 361–365. https://doi.org/10.1037/0033-2909.111.2.361

Miller, J. J. (1978). The inverse of the Freeman-Tukey double arcsine transformation. *American Statistician*, **32**(4), 138. https://doi.org/10.1080/00031305.1978.10479283

Rosenthal, R., & Rubin, D. B. (1982). A simple, general purpose display of magnitude of experimental effect. *Journal of Educational Psychology*, **74**(2), 166–169. https://doi.org/10.1037/0022-0663.74.2.166

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit random-effects model
res <- rma(yi, vi, data=dat)

### average risk ratio with 95% CI (but technically, this provides an
### estimate of the median risk ratio, not the mean risk ratio!)
predict(res, transf=exp)

### average risk ratio with 95% CI using the integral transformation
predict(res, transf=transf.exp.int, targs=list(tau2=res$tau2, lower=-4, upper=4))
```

trimfill *Trim and Fill Analysis for 'rma.uni' Objects*

Description

Carry out a trim and fill analysis for objects of class "rma.uni".

Usage

```
trimfill(x, ...)

## S3 method for class 'rma.uni'
trimfill(x, side, estimator="L0", maxiter=100, verbose=FALSE, ilim, ...)
```

Arguments

x	an object of class "rma.uni".
side	optional character string (either "left" or "right") to specify on which side of the funnel plot the missing studies should be imputed. If left unspecified, the side is chosen within the function depending on the results of Egger's regression test (see regtest for details on this test).
estimator	character string (either "L0", "R0", or "Q0") to specify the estimator to use for estimating the number of missing studies (the default is "L0").
maxiter	integer to specify the maximum number of iterations to use for the trim and fill method (the default is 100).
verbose	logical to specify whether output should be generated on the progress of the iterative algorithm used as part of the trim and fill method (the default is FALSE).
ilim	limits for the imputed values. If unspecified, no limits are used.
...	other arguments.

Details

The trim and fill method is a nonparametric (rank-based) data augmentation technique proposed by Duval and Tweedie (2000a, 2000b; see also Duval, 2005). The method can be used to estimate the number of studies missing from a meta-analysis due to suppression of the most extreme results on one side of the funnel plot. The method then augments the observed data so that the funnel plot is more symmetric and recomputes the summary estimate based on the complete data. The trim and fill method can only be used in the context of a fixed- or random-effects model (i.e., in models without moderators). The method should not be regarded as a way of yielding a more 'valid' estimate of the overall effect or outcome, but as a way of examining the sensitivity of the results to one particular selection mechanism (i.e., one particular form of publication bias).

Value

An object of class `c("rma.uni.trimfill","rma.uni","rma")`. The object is a list containing the same components as objects created by `rma.uni`, except that the data are augmented by the trim and fill method. The following components are also added:

`k0`	estimated number of missing studies.
`side`	either `"left"` or `"right"`, indicating on which side of the funnel plot the missing studies (if any) were imputed.
`se.k0`	standard error of k0.
`p.k0`	p-value for the test of H_0: no missing studies on the chosen side (only when `estimator="R0"`; `NA` otherwise).
`yi`	the observed effect sizes or outcomes plus the augmented values (if there are any).
`vi`	the corresponding sampling variances
`fill`	a logical vector indicating which of the values in `yi` are the observed (`FALSE`) and the augmented (`TRUE`) data.

The results of the fitted model after the data augmentation are printed with the `print.rma.uni` function. Calling `funnel.rma` on the object provides a funnel plot of the observed and augmented data.

Note

Three different estimators for the number of missing studies were proposed by Duval and Tweedie (2000a, 2000b). Based on these articles and Duval (2005), `"R0"` and `"L0"` are recommended. An advantage of estimator `"R0"` is that it provides a test of the null hypothesis that the number of missing studies (on the chosen side) is zero.

If the outcome measure used for the analysis is bounded (e.g., correlations are bounded between -1 and +1, proportions are bounded between 0 and 1), one can use the `ilim` argument to enforce those limits when imputing values (imputed values cannot exceed those bounds then).

The model used during the trim and fill procedure is the same as used by the original model object. Hence, if a fixed-effects model is passed to the function, then a fixed-effects model is also used during the trim and fill procedure and the results provided are also based on a fixed-effects model. This would be a ‘fixed-fixed’ approach. Similarly, if a random-effects model is passed to the function, then the same model is used as part of the trim and fill procedure and for the final analysis. This would be a ‘random-random’ approach. However, one can also easily fit a different model for the final analysis than was used for the trim and fill procedure. See ‘Examples’ for an illustration of a ‘fixed-random’ approach.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Duval, S. J., & Tweedie, R. L. (2000a). Trim and fill: A simple funnel-plot-based method of testing and adjusting for publication bias in meta-analysis. *Biometrics*, **56**(2), 455–463. https://doi.org/10.1111/j.0006-341x.2000.00455.x

Duval, S. J., & Tweedie, R. L. (2000b). A nonparametric "trim and fill" method of accounting for publication bias in meta-analysis. *Journal of the American Statistical Association*, **95**(449), 89–98. https://doi.org/10.1080/01621459.2000.10473905

Duval, S. J. (2005). The trim and fill method. In H. R. Rothstein, A. J. Sutton, & M. Borenstein (Eds.) *Publication bias in meta-analysis: Prevention, assessment, and adjustments* (pp. 127–144). Chichester, England: Wiley.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

`funnel.rma`

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### meta-analysis of the log risk ratios using a fixed-effects model
res <- rma(yi, vi, data=dat, method="FE")
res.tf <- trimfill(res)
res.tf
funnel(res.tf, legend=TRUE, cex=1.2)

### estimator "R0" also provides test
res.tf <- trimfill(res, estimator="R0")
res.tf

### meta-analysis of the log risk ratios using a random-effects model
res <- rma(yi, vi, data=dat)
res.tf <- trimfill(res)
res.tf
funnel(res.tf, legend=TRUE, cex=1.2)

### the examples above are fixed-fixed and random-random approaches

### illustration of a fixed-random approach
res <- rma(yi, vi, data=dat, method="FE")
res.tf <- trimfill(res)
filled <- data.frame(yi = res.tf$yi, vi = res.tf$vi, fill = res.tf$fill)
filled
rma(yi, vi, data=filled)
```

update.rma *Model Updating for 'rma' Objects*

Description

The function can be used to update and (by default) re-fit "rma" models. It does this by extracting the call stored in the object, updating the call and (by default) evaluating that call.

Usage

```
## S3 method for class 'rma'
update(object, formula., ..., evaluate = TRUE)
```

Arguments

object	an object of class "rma".
formula.	changes to the formula. See 'Details'.
...	additional arguments to the call, or arguments with changed values.
evaluate	logical to specify whether to evaluate the new call or just return the call.

Details

For objects of class "rma.uni", "rma.glmm", and "rma.mv", the formula. argument can be used to update the set of moderators included in the model (see 'Examples').

Value

If evaluate=TRUE the fitted object, otherwise the updated call.

Author(s)

The present function is based on update.default, with changes made by Wolfgang Viechtbauer (<wvb@metafor-project.org>) so that the formula updating works with the (somewhat non-standard) interface of the rma.uni, rma.glmm, and rma.mv functions.

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mh, rma.peto, rma.glmm, rma.mv

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit random-effects model (method="REML" is default)
res <- rma(yi, vi, data=dat, digits=3)
res

### fit mixed-effects model with two moderators (absolute latitude and publication year)
res <- update(res, ~ ablat + year)
res

### remove 'year' moderator
res <- update(res, ~ . - year)
res

### fit model with ML estimation
update(res, method="ML")

### example with rma.glmm()
res <- rma.glmm(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, digits=3)
res <- update(res, mods = ~ ablat)
res

### fit conditional model with approximate likelihood
update(res, model="CM.AL")
```

vcov.rma — *Extract Various Types of Variance-Covariance Matrices from ’rma’ Objects*

Description

The function extracts various types of variance-covariance matrices from objects of class "rma". By default, the variance-covariance matrix of the parameter estimates (fixed effects) is returned.

Usage

```
## S3 method for class 'rma'
vcov(object, type="fixed", ...)
```

Arguments

`object` an object of class "rma".

`type` character string to specify the type of variance-covariance matrix to return: type="fixed" returns the variance-covariance matrix of the fixed effects (the default), type="obs" returns the marginal variance-covariance matrix of the observed effect sizes or outcomes, type="fitted" returns the variance-covariance matrix of the fitted values, type="resid" returns the variance-covariance matrix of the residuals.

... other arguments.

Details

Note that type="obs" currently only works for object of class "rma.uni" and "rma.mv".

For objects of class "rma.uni", the marginal variance-covariance matrix of the observed effect sizes or outcomes is just a diagonal matrix with $\hat{\tau}^2 + v_i$ along the diagonal, where $\hat{\tau}^2$ is the estimated amount of (residual) heterogeneity (set to 0 in fixed-effects models) and v_i is the sampling variance of the ith study.

For objects of class "rma.mv", the structure can be more complex and depends on the random effects included in the model.

Value

A matrix corresponding to the requested variance-covariance matrix.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mh, rma.peto, rma.glmm, rma.mv

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)

### var-cov matrix of the fixed effects (i.e., the model coefficients)
vcov(res)

### marginal var-cov matrix of the observed log risk ratios
vcov(res, type="obs")

### var-cov matrix of the fitted values
vcov(res, type="fitted")

### var-cov matrix of the residuals
vcov(res, type="resid")
```

vec2mat *Convert a Vector into a Square Matrix*

Description

Function to convert a vector into a square matrix by filling up the lower triangular part of the matrix.

Usage

```
vec2mat(x, diag=FALSE, corr=!diag, dimnames)
```

Arguments

`x`	a vector of the correct length.
`diag`	logical to specify whether the vector also contains the diagonal values of the lower triangular part of the matrix (the default is `FALSE`).
`corr`	logical to specify whether the diagonal of the matrix should be replaced with 1's (the default is to do this when `diag=FALSE`).
`dimnames`	optional vector of the correct length with the dimension names of the matrix.

Details

The values in `x` are filled into the lower triangular part of a square matrix with the appropriate dimensions (which are determined based on the length of `x`). If `diag=TRUE`, then `x` is assumed to also contain the diagonal values of the lower triangular part of the matrix. If `corr=TRUE`, then the diagonal of the matrix is replaced with 1's.

Value

A matrix.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

Examples

```
vec2mat(1:6, corr=FALSE)
vec2mat(seq(0.2, 0.7, by=0.1), corr=TRUE)
vec2mat(1:10, diag=TRUE)
vec2mat(1:6, corr=FALSE, dimnames=c("A","B","C","D"))
```

vif *Variance Inflation Factors for 'rma' Objects*

Description

Compute variance inflation factors (VIFs) for objects of class `"rma"`.

Usage

```
vif(x, ...)

## S3 method for class 'rma'
vif(x, btt, intercept=FALSE, table=FALSE, digits, ...)

## S3 method for class 'vif.rma'
print(x, digits=x$digits, ...)
```

Arguments

`x` an object of class `"rma"` (for `vif`) or `"vif.rma"` (for `print`).

`btt` optional vector of indices to specify a set of coefficients for which a generalized variance inflation factor (GVIF) should be computed. Can also be a string to grep for. See 'Details'.

`intercept` logical to specify whether to include the intercept (if the model includes one) in the computation of the VIFs (the default is `FALSE`). See 'Note'.

`table` logical to specify whether the VIFs should be added to the model coefficient table (the default is `FALSE`).

`digits` integer to specify the number of decimal places to which the printed results should be rounded. If unspecified, the default is to take the value from the object.

`...` other arguments.

Details

The function computes variance inflation factors (VIFs) for meta-regression models. Hence, the model specified via argument `x` must include moderator variables (and more than one for this to be useful, as the VIF for a model with a single moderator variable will always be equal to 1).

Let b_j denote the estimate of the jth model coefficient of a particular meta-regression model and $\text{Var}[b_j]$ its variance (i.e., the corresponding diagonal element from the matrix obtained with the `vcov.rma` function). Moreover, let b'_j denote the estimate of the same model coefficient if the other moderator variables in the model had *not* been included in the model and $\text{Var}[b'_j]$ the corresponding variance. Then the VIF for the model coefficient is given by

$$\text{VIF}[b_j] = \frac{\text{Var}[b_j]}{\text{Var}[b'_j]},$$

which indicates the inflation in the variance of the estimated model coefficient due to potential collinearity of the jth moderator variable with the other moderator variables in the model. Taking the square root of a VIF gives the corresponding standard error inflation factor (SIF).

If `btt` is not specified, then the VIF is computed for each individual model coefficient. However, if the model includes factors (coded in terms of multiple dummy variables) or other sets of moderator variables that belong together (e.g., for polynomials or cubic splines), one may want to examine how much the variance in all of the coefficients in the set is jointly impacted by collinearity with the other moderator variables in the model. For this, we can compute a generalized variance inflation factor (GVIF) (Fox & Monette, 1992) by setting the `btt` argument equal to the indices of those coefficients for which the GVIF should be computed. The square root of a GVIF indicates the inflation in the confidence ellipse/(hyper)ellipsoid for the set of coefficients corresponding to the set due to collinearity. However, to make this value more directly comparable to SIFs (based on single coefficients) or when the set includes a different number of coefficients, the function computes the generalized standard error inflation factor (GSIF) by raising the GVIF to the power of $1/(2m)$ (where m denotes the number of coefficients in the set).

Value

If `btt` is not specified, either a vector (if `table=FALSE`) with the VIFs or a data frame (if `table=TRUE`) with the following elements:

`estimate`	estimated model coefficients.
`se`	corresponding standard errors.
`zval`	corresponding test statistics.
`pval`	corresponding p-values.
`ci.lb`	corresponding lower bound of the confidence intervals.
`ci.ub`	corresponding upper bound of the confidence intervals.
`vif`	corresponding variance inflation factors.
`sif`	corresponding standard error inflation factors.

If `btt` is specified, a list with elements `gvif` and `gsif` with the GVIF and GSIF values for the set of coefficients specified.

Note

The values of the (G)VIFs are invariant to the scaling of the predictor variables if the model includes an intercept that is removed when inverting the correlation matrix of the model coefficients to compute the (G)VIFs. This is the default behavior. See ‘Examples’.

Author(s)

Wolfgang Viechtbauer `<wvb@metafor-project.org>` `https://www.metafor-project.org`

References

Belsley, D. A., Kuh, E., & Welsch, R. E. (1980). *Regression diagnostics*. New York: Wiley.

Fox, J., & Monette, G. (1992). Generalized collinearity diagnostics. *Journal of the American Statistical Association*, **87**(417), 178-183. https://doi.org/10.2307/2290467

Stewart, G. W. (1987). Collinearity and least squares regression. *Statistical Science*, **2**(1), 68-84. https://doi.org/10.1214/ss/1177013439

Wax, Y. (1992). Collinearity diagnosis for a relative risk regression-analysis: An application to assessment of diet cancer relationship in epidemiologic studies. *Statistics in Medicine*, **11**(10), 1273–1287. https://doi.org/10.1002/sim.4780111003

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mv, rma.glmm

Examples

```
### copy data from Bangert-Drowns et al. (2004) into 'dat'
dat <- dat.bangertdrowns2004

### fit mixed-effects meta-regression model
res <- rma(yi, vi, mods = ~ length + wic + feedback + info + pers + imag + meta, data=dat)

### get variance inflation factors
vif(res)

### show that VIFs are not influenced by scaling of the predictors
u <- scale # to standardize the predictors
res <- rma(yi, vi, mods = ~ u(length) + u(wic) + u(feedback) + u(info) +
                            u(pers) + u(imag) + u(meta), data=dat)
vif(res)

### get full table
vif(res, table=TRUE)

### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit meta-regression model where one predictor (alloc) is a three-level factor
res <- rma(yi, vi, mods = ~ ablat + alloc, data=dat)

### get variance inflation factors for all individual coefficients
vif(res, table=TRUE)

### generalized variance inflation factor for the 'alloc' factor
vif(res, btt=3:4)

### can also specify a string to grep for
vif(res, btt="alloc")
```

weights.rma *Compute Weights for 'rma' Objects*

Description

The function computes the weights given to the observed effect sizes or outcomes during the model fitting for objects of class "rma.uni", "rma.mh", "rma.peto", and "rma.mv".

Usage

```
## S3 method for class 'rma.uni'
weights(object, type="diagonal", ...)
## S3 method for class 'rma.mh'
weights(object, type="diagonal", ...)
## S3 method for class 'rma.peto'
weights(object, type="diagonal", ...)
## S3 method for class 'rma.glmm'
weights(object, ...)
## S3 method for class 'rma.mv'
weights(object, type="diagonal", ...)
```

Arguments

`object`	an object of class "rma.uni", "rma.mh", "rma.peto", or "rma.mv". The method is not yet implemented for objects of class "rma.glmm".
`type`	character string to specify whether to return only the diagonal of the weight matrix ("diagonal") or the entire weight matrix ("matrix"). For "rma.mv", this can also be "rowsum" for 'row-sum weights' (for intercept-only models).
`...`	other arguments.

Value

Either a vector with the diagonal elements of the weight matrix or the entire weight matrix. When only the diagonal elements are returned, they are given in % (and they add up to 100%).

When the entire weight matrix is requested, this is always a diagonal matrix for objects of class "rma.uni", "rma.mh", "rma.peto".

For "rma.mv", the structure of the weight matrix depends on the model fitted (i.e., the random effects included and the variance-covariance matrix of the sampling errors) but is often more complex and not just diagonal.

For "rma.mv" intercept-only models, one can also take the sum over the rows in the weight matrix, which are actually the weights assigned to the observed effect sizes or outcomes when estimating the model intercept. These weights can be obtained with type="rowsum" (as with type="diagonal", they are also given in %).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org> https://www.metafor-project.org

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. https://doi.org/10.18637/jss.v036.i03

See Also

rma.uni, rma.mh, rma.peto, rma.mv, influence.rma.uni

Examples

```
### calculate log risk ratios and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### fit mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods = ~ ablat + year, data=dat)

### extract the model fitting weights (in %)
weights(res)

### extract the weight matrix
weights(res, type="matrix")
```

Index