

Package ‘metafor’

February 9, 2012

Version 1.6-0

Date 2011-04-12

Title Meta-Analysis Package for R

Author Wolfgang Viechtbauer <wvb@metafor-project.org>

Maintainer Wolfgang Viechtbauer <wvb@metafor-project.org>

Description The metafor package consists of a collection of functions for conducting meta-analyses in R. The package includes functions to calculate various effect size 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, and L’Abbe plots). For meta-analyses of 2x2 table data, the Mantel-Haenszel and Peto’s method are also implemented.

Depends R (>= 2.12.2), stats, nlme, Formula

License GPL-2

URL <http://www.metafor-project.org/>

Repository CRAN

Date/Publication 2011-04-13 09:25:54

R topics documented:

metafor-package	3
addpoly	6
addpoly.default	7
addpoly.rma	10
anova.rma.uni	12
blup	14
blup.rma.uni	15
coef.permutest.rma.uni	17
coef.rma	18
confint.rma.uni	20

cumul	22
cumul.rma.mh	23
cumul.rma.uni	25
dat.bcg	27
dat.co2	29
dat.empint	31
dat.los	32
dat.warfarin	34
escalc	35
fitstats	43
fitstats.rma	44
fitted.rma	46
forest	47
forest.cumul.rma	48
forest.default	51
forest.rma	54
fsn	59
funnel	61
funnel.rma	62
hatvalues.rma.uni	65
influence.rma.uni	66
labbe	68
labbe.rma	69
leave1out	71
leave1out.rma.mh	72
leave1out.rma.uni	74
metafor.news	75
permutest	76
permutest.rma.uni	77
plot.infl.rma.uni	80
plot.rma.uni	82
predict.rma	83
print.anova.rma.uni	86
print.fsn	87
print.infl.rma.uni	88
print.list.rma	89
print.permutest.rma.uni	90
print.predict.rma	91
print.ranktest.rma	92
print.regtest.rma	93
print.rma.uni	94
qqnorm.rma.uni	95
radial	97
radial.rma	98
ranktest	100
ranktest.rma	101
regtest	103
regtest.rma	104

residuals.rma	107
rma.mh	109
rma.peto	113
rma.uni	116
transf	124
trimfill	126
trimfill.rma.uni	127
vcov.rma	129
weights.rma.uni	130

Index	132
--------------	------------

metafor-package	<i>Metafor: A Meta-Analysis Package for R</i>
-----------------	---

Description

The **metafor** package consists of a collection of functions for conducting meta-analyses in R. The package includes functions for calculating various effect size or outcome measures frequently used in meta-analyses (e.g., risk differences, risk ratios, odds ratios, standardized mean differences, Fisher’s z-transformed correlation coefficients) and then allows the user to fit fixed-, random-, and mixed-effects models to these data. By including study-level covariates (‘moderators’) in these models, so-called ‘meta-regression’ analyses can be carried out. For meta-analyses of 2x2 table data, the Mantel-Haenszel and Peto’s method are also implemented.

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

The package also provides functions for creating forest, funnel, radial, normal quantile-quantile, and L’Abbe plots. The presence of publication bias (or more accurately, for funnel plot asymmetry) and its potential impact on the results can be examined via the rank correlation and Egger’s regression test and by applying the trim and fill method.

The **rma.uni** Function

The various meta-analytic models that are usually used in practice are special cases of the 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., log odds ratios, log relative risks, risk differences, mean differences, standardized mean differences, raw correlation coefficients, correlation coefficients transformed with Fisher’s r-to-z transformation, and so on). For details on these effect size or outcome measures, see the documentation of 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 i^{th} study. Let θ_i denote the corresponding (unknown) true effect or outcome in the i^{th} study, such that $y_i|\theta_i \sim N(\theta_i, v_i)$. In other words, the observed effects or

outcomes are assumed to be unbiased and normally distributed estimates of the corresponding true effects or 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 ratios, Fisher's r-to-z transformation for correlations, the bias correction for standardized mean differences described by Hedges & Olkin, 1985; see section [escalc](#) for more details).

The **fixed-effects model** conditions on the true effects or outcomes and therefore provides a *conditional inference* about the set of k studies included in the meta-analysis. 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 in the set of 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 in the set of k studies (i.e., an estimate of $\bar{\theta}_u = \sum_{i=1}^k \theta_i / k$).

Moderators can be included in the fixed-effects model, yielding a **fixed-effects with moderators model**. Again, since the model conditions on the set of k studies included in the meta-analysis, the regression coefficients from the fitted model estimate the weighted least squares relationship between the true effects and the moderator variables within the set of k studies included in the meta-analysis (again using weights equal to $w_i = 1/v_i$). The (unweighted) least squares relationship between the true effects and the moderator variables can be obtained when using the unweighted estimation method.

The **random-effects model** does not condition on the true effects. Instead, the k studies included in the meta-analysis are assumed to be a random selection from a hypothetical population of studies. One can envision this hypothetical population as 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 $\theta_i \sim N(\mu, \tau^2)$, that is, the true effects or outcomes in the population of studies are assumed to be normally distributed with μ denoting the average effect and τ^2 denoting the variance of the true effects in the population (τ^2 is therefore often referred to as the 'amount of heterogeneity' in the true effects). The fitted model provides an estimate of μ and τ^2 . Consequently, the random-effects model provides an *unconditional inference* about the average effect in the population of studies from which the k studies included in the meta-analysis are assumed to be a random selection.

When including moderator variables in the random-effects model, we obtain what is typically called a **mixed-effects model** in the meta-analytic literature. The coefficients from the fitted model then estimate the relationship between the average true effect or outcome in the population of studies and the moderator variables included in the model. The value of τ^2 in the mixed-effects model denotes the 'amount of residual heterogeneity' in the true effects or outcomes (i.e., the amount of variability among the true effects or outcomes that is not accounted for by the moderators included in the model).

One can also choose between weighted and unweighted estimation in the context of the random- and mixed-effects model, although the parameters that are estimated remain the same regardless of the estimation method used (as opposed to the fixed-effects model case, where the parameter estimated is different for weighted and unweighted estimation).

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 or outcomes are homogeneous (i.e., that θ_i is equal to some common value θ for 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 (more specifically, to sets of k studies with true effects equal to the true effects of the k studies included in the meta-analysis). On the other hand, the random-effects model provides an inference about the average effect in the entire population of studies from which the included studies are assumed to be a random selection.

In the special case that the true effects are actually homogeneous, 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 are really homogeneous or not, a researcher should decide on the type of inference desired before examining the data and choose the model accordingly. For more details on the distinction between fixed- and random-effects models, see Hedges and Vevea (1998) and Laird and Mosteller (1990).

The `rma.mh` Function

The Mantel-Haenszel method provides an alternative approach to fitting the fixed-effects model when dealing with studies providing data in the form of 2x2 tables or in the form of incidence rates (Mantel & Haenszel, 1959). The method is particularly advantageous when aggregating a large number of studies/tables 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 odds ratios, relative risks, risk differences, and incidence rate ratios. The Mantel-Haenszel method is always based on a weighted estimation approach.

The `rma.peto` Function

Yet another method that can be used in the context of a meta-analysis of 2x2 tables is Peto's method (see Yusuf et al., 1985), implemented in the `rma.peto` function. It is a weighted estimation approach for the combination of odds ratios.

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 very welcome.

And since this is a frequently-asked-question: Functions for more complex syntheses (e.g., network meta-analyses, multivariate meta-analyses, meta-analyses with correlated outcomes) are currently under development and will be incorporated into the package in the future.

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. <http://www.jstatsoft.org/v36/i03/>.

Getting Started with the Package

The paper mentioned above is a good starting place for those interesting in using the `metafor` 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 pretty thorough introduction. The paper can be freely downloaded from the URL given above.

Alternatively (or 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` and `rma.peto` functions if you intend to use the Mantel-Haenszel or Peto's method). The model fitting functions (i.e., `rma.uni`, `rma.mh`, and `rma.peto`) provide links to the help pages of many additional functions, which can be used after fitting a model.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

Please post any questions about the package to the R-help mailing list (<https://stat.ethz.ch/mailman/listinfo/r-help>). Make sure you first read and follow the posting guide (<http://www.r-project.org/posting-guide.html>). To make sure that my mail filter catches your post, please include the package name (i.e., 'metafor') in your post.

References

- Cooper, H. C., Hedges, L. V., & Valentine, J. C. (Eds.) (2009). *The handbook of research synthesis and meta-analysis* (2nd ed.). 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**, 486–504.
- Laird, N. M. & Mosteller, F. (1990). Some statistical methods for combining experimental results. *International Journal of Technology Assessment in Health Care*, **6**, 5–30.
- Mantel, N. & Haenszel, W. (1959). Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the National Cancer Institute*, **22**, 719–748.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/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**, 335–371.

addpoly

Add Polygons to Forest Plot

Description

The function `addpoly` is generic. It can be used to add polygons to a forest plot, for example, to indicate summary estimates and fitted/predicted values.

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`, or `rma.peto` 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 is used to add a polygon to the bottom of an existing forest plot, showing the summary estimate with corresponding 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>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[addpoly.rma](#), [addpoly.default](#), [forest.rma](#), [forest.default](#)

addpoly.default *Add Polygons to Forest Plot*

Description

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

Usage

```
## Default S3 method:  
addpoly(x, vi, sei, ci.lb, ci.ub,  
        rows=-1, level=95, digits=2, annotate=TRUE,  
        mlab=NULL, transf=FALSE, atranf=FALSE, targs,  
        col="black", efac=1, cex=NULL, ...)
```

Arguments

<code>x</code>	a vector with the values at which the polygons should be drawn.
<code>vi</code>	a vector with the corresponding variances.
<code>sei</code>	a vector with the corresponding standard errors. (note: only one of the two, <code>vi</code> or <code>sei</code> , needs to be specified)
<code>ci.lb</code>	a vector with the corresponding lower confidence intervals bounds. Not needed if <code>vi</code> or <code>sei</code> is specified. See 'Details'.
<code>ci.ub</code>	a vector with the corresponding upper confidence intervals bounds. Not needed if <code>vi</code> or <code>sei</code> is specified. See 'Details'.
<code>rows</code>	a vector specifying the rows (or more generally, the horizontal positions) for plotting the polygons (defaults is -1). Can also be a single value specifying the row (horizontal position) of the first polygon (the remaining polygons are then plotted below this starting row).
<code>level</code>	a numerical value between 0 and 100 specifying the confidence interval level (default is 95).
<code>digits</code>	integer value specifying the number of decimal places to which the annotations should be rounded (default is 2).
<code>annotate</code>	logical specifying whether annotations should be added to the plot for the polygons that are drawn (default is TRUE).
<code>mlab</code>	optional character vector with the same length as <code>x</code> giving labels for the polygons that are drawn. Defaults to NULL, which suppresses labels.
<code>transf</code>	an optional argument specifying the name of a function that should be used to transform the <code>x</code> values and confidence interval bounds. Defaults to FALSE, which means that no transformation is used.
<code>atransf</code>	an optional argument specifying the name of a function that should be used to transform the annotations. Defaults to FALSE, which means that no transformation is used.
<code>targs</code>	optional arguments needed by the function specified via <code>transf</code> or <code>atransf</code> .
<code>col</code>	color of the polygons that are drawn (default is "black").
<code>efac</code>	vertical expansion factor for the polygons. The default value of 1 should usually work okay.
<code>cex</code>	an optional symbol expansion factor. If NULL (default), the function tries to set this to a sensible value.
<code>...</code>	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 subgrouping of the studies or from models involving moderators can be added to the plot this way. See examples below.

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).

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>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[forest.rma](#), [forest.default](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a mixed-
### effects model with absolute latitude as a moderator
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat),
          slab=paste(author, year, sep=" ", data=dat.bcg,
                    measure="RR", method="REML"))

### forest plot of the observed relative risks
forest(res, addfit=FALSE, atranf=exp, ylim=c(-3.5,16))

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

### add predicted average relative risks to forest plot
addpoly(x$pred, sei=x$se, atranf=exp,
        mlab=c("10 Degrees", "30 Degrees", "50 Degrees"))
abline(h=0)

### forest plot with subgrouping of studies and summaries per subgroup
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, measure="RR",
          slab=paste(author, year, sep=" ", method="REML"))
forest(res, slab=paste(dat.bcg$author, dat.bcg$year, sep=" "),
      xlim=c(-16, 6), at=log(c(.05, .25, 1, 4)), atranf=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=order(dat.bcg$alloc), rows=c(3:4,9:15,20:23),
      mlab="RE Model for All Studies")
op <- par(cex=.75, font=4)
```

```

text(-16, c(24,16,5), c("Systematic Allocation", "Random Allocation",
                        "Alternate Allocation"), pos=4)
par(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"))
text(-16, 26, "Author(s) and Year", pos=4)
text(6, 26, "Relative Risk [95% CI]", pos=2)
par(op)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, measure="RR",
           subset=(alloc=="systematic"), method="REML")
addpoly(res, row=18.5, cex=.75, atranf=exp, mlab="RE Model for Subgroup")
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, measure="RR",
           subset=(alloc=="random"), method="REML")
addpoly(res, row=7.5, cex=.75, atranf=exp, mlab="RE Model for Subgroup")
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, measure="RR",
           subset=(alloc=="alternate"), method="REML")
addpoly(res, row=1.5, cex=.75, atranf=exp, mlab="RE Model for Subgroup")

```

addpoly.rma

Add Summary Estimate Polygon to Forest Plot

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, digits=2, annotate=TRUE,
        mlab=NULL, transf=FALSE, atranf=FALSE, targs,
        col="black", efac=1, cex=NULL, ...)

```

Arguments

x	an object of class "rma".
row	value specifying the row (or more generally, the horizontal position) for plotting the polygon (default is -2).
level	a numerical value between 0 and 100 specifying the confidence interval level (the default is to take the value from the object).
digits	integer value specifying the number of decimal places to which the annotations should be rounded (default is 2).
annotate	logical specifying whether annotations for the summary estimate should be added to the plot (default is TRUE).
mlab	optional character string giving a label for the summary estimate polygon. Defaults to NULL, which means that the label is created within the function.

transf	an optional argument specifying the name of a function that should be used to transform the summary estimate and confidence interval bound. Defaults to FALSE, which means that no transformation is used.
atransf	an optional argument specifying the name of a function that should be used to transform the annotations. Defaults to FALSE, which means that no transformation is used.
targs	optional arguments needed by the function specified via transf or atransf.
col	color of the polygon that is drawn (default is "black").
efac	vertical expansion factor for the polygon. The default value of 1 should usually work okay.
cex	an optional symbol expansion factor. If NULL (default), the function tries to set this to a sensible value.
...	other arguments.

Details

The function can be used to add a polygon to an existing forest plot created with the `forest` function. The polygon shows the summary estimate based on a fixed- or random-effects model.

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>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

`forest.rma`, `forest.default`

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

### forest plot of the observed relative risks with summary estimate
forest(res, atransf=exp, ylim=c(-2.5,16))
```

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

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

 anova.rma.uni

Compare Fit Statistics and Likelihoods of rma.uni Objects

Description

The function provides a full versus reduced model comparison of two objects of class "rma.uni". Model fit statistics for the two models are provided. A likelihood ratio test comparing the two models is also performed.

Usage

```
## S3 method for class 'rma.uni'
anova(object, object2, digits=object$digits, ...)
```

Arguments

object	an object of class "rma.uni".
object2	an object of class "rma.uni".
digits	an integer specifying 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 two models must be based on the same set of data and should be nested for the likelihood ratio test to make sense.

Value

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

fit.stats.f	log likelihood, deviance, AIC, and BIC for the full model.
fit.stats.r	log likelihood, deviance, AIC, and BIC for the reduced model.
p.f	number of parameters in the full model.
p.r	number of parameters in the reduced model.
LRT	likelihood ratio test statistic.
pval	p-value for the likelihood ratio test.
QE.f	test statistic for the test of (residual) heterogeneity from the full model.

QE.r	test statistic for the test of (residual) heterogeneity from the reduced model.
tau2.f	tau2 value from the full model.
tau2.r	tau2 value from the reduced model.
VAf	amount of (residual) heterogeneity in the reduced model that is accounted for in the full model. NA for fixed-effects models or if the amount of heterogeneity in the reduced model is equal to zero.

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

Note

Note that likelihood ratio tests are not meaningful when using REML estimation and the two models have different fixed effects.

In principle, one can also consider likelihood ratio tests for (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 from that 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. Moreover, the Q-test usually keeps better control of the Type I error rate and therefore should be preferred (see Viechtbauer, 2007, for more details).

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Viechtbauer, W. (2007). Hypothesis tests for population heterogeneity in meta-analysis. *British Journal of Mathematical and Statistical Psychology*, **60**, 29–60.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

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

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

### random-effects model
res2 <- rma(yi, vi, data=dat, method="ML")
```

```
### mixed-effects model with two moderators (absolute latitude and publication year)
res1 <- rma(yi, vi, mods=cbind(ablat, year), data=dat, method="ML")

anova(res1, res2)
```

blup

Best Linear Unbiased Predictions

Description

The function `blup` is generic. It extracts the best linear unbiased predictions (BLUPs) for specific classes of objects. BLUPs combine the fitted values based on the fixed effects in the model and the estimated contributions of the random effects.

Usage

```
blup(x, ...)
```

Arguments

`x` an object for which BLUPs are meaningful. See ‘Details’.
`...` other arguments.

Details

Currently, there is only a specific method for objects of class `"rma.uni"` created by the `rma.uni` function. Accordingly, the corresponding method is called `blup.rma.uni`. See the documentation for that function for more details.

Value

Best linear unbiased predictions and possibly corresponding standard errors and prediction interval bounds.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Robinson, G. K. (1991). That BLUP is a good thing: The estimation of random effects. *Statistical Science*, **6**, 15–32.

See Also

[blup.rma.uni](#)

Description

The function calculates the best linear unbiased predictions (BLUPs) of the true 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

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

Arguments

x	an object of class "rma.uni".
level	a numerical value between 0 and 100 specifying the prediction interval level (the default is to take the value from the object).
digits	an integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
transf	an optional argument specifying the name of a function that should be used to transform the predicted values and interval bounds (e.g., transf=exp). Defaults to FALSE, which means that 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 "list.rma" object is formatted and printed with `print.list.rma`.

Note

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 automatically be equal to the usual 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 set equal to NA.

The normal distribution is used to calculate the prediction intervals. When the model was fitted with the Knapp and Hartung (2003) method (i.e., `knha=TRUE` in the [rma.uni](#) function), then the 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>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

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**, 1249–1261.
- Raudenbush, S. W. & Bryk, A. S. (1985). Empirical Bayes meta-analysis. *Journal of Educational Statistics*, **10**, 75–98.
- Robinson, G. K. (1991). That BLUP is a good thing: The estimation of random effects. *Statistical Science*, **6**, 15–32.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.uni](#), [predict.rma](#), [fitted.rma](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

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

```

blup(res, transf=exp)

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

```

coef.permutest.rma.uni

Model Coefficients for permutest.rma.uni Objects

Description

The function extracts the estimated model coefficients, corresponding standard errors, test statistics, p-values (based on the permutation test), 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.
ci.lb	lower bound of the confidence interval(s).
ci.ub	upper bound of the confidence interval(s).

When the model was fitted with the Knapp and Hartung (2003) method (i.e., knha=TRUE in the [rma.uni](#) function), then zval is called tval in the data frame that is returned by the function.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[permutest.rma.uni](#), [rma.uni](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a mixed-effects model
### with two moderators (absolute latitude and publication year)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat, year),
          data=dat.bcg, measure="RR", method="REML")

### permutation test
## Not run: pres <- permutest(res)
## Not run: coef(pres)
```

 coef.rma

Model Coefficients for rma Objects

Description

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

Usage

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

Arguments

object an object of class "rma".
 ... 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	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 the Knapp and Hartung (2003) method (i.e., knha=TRUE in the [rma.uni](#) function), then zval is called tval in the data frame that is returned by the function.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.uni](#), [rma.mh](#), [rma.peto](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a mixed-effects model
### with two moderators (absolute latitude and publication year)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat, year),
          data=dat.bcg, measure="RR", method="REML")
coef(res)
```

Description

The function calculates a confidence interval for the amount of (residual) heterogeneity in random- and mixed-effects models that were fitted with the `rma.uni` function. Confidence intervals for the model coefficients can also be obtained.

Usage

```
## S3 method for class 'rma.uni'
confint(object, parm, level=object$level, fixed=FALSE,
        random=TRUE, digits=object$digits, control, ...)
## S3 method for class 'rma.mh'
confint(object, parm, level=object$level, digits=object$digits, ...)
## S3 method for class 'rma.peto'
confint(object, parm, level=object$level, digits=object$digits, ...)
```

Arguments

<code>object</code>	an object of class "rma.uni", "rma.mh", or "rma.peto".
<code>parm</code>	this argument is here for compatability with the generic function <code>confint</code> , but is ignored.
<code>level</code>	a numerical value between 0 and 100 specifying the confidence interval level (the default is to take the value from the object).
<code>fixed</code>	logical indicating whether confidence intervals for the model coefficients should be returned (default is FALSE).
<code>random</code>	logical indicating whether a confidence interval for the amount of (residual) heterogeneity should be returned (default is TRUE).
<code>digits</code>	an integer specifying the number of decimal places to which the results should be rounded (the default is to take the value from the object).
<code>control</code>	a list of control values for the iterative algorithm. Defaults to an empty list, which means that default values are defined inside the function. See 'Note'.
<code>...</code>	other arguments.

Details

The confidence intervals for the model coefficients are simply the usual Wald-type intervals which are also shown when printing the fitted object.

The confidence interval for the amount of (residual) heterogeneity is obtained iteratively via the Q-profile method as described by Hartung and Knapp (2005) and Viechtbauer (2007). The 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. For random-effects models, confidence intervals for I^2 and H^2 are also provided (Higgins & Thompson, 2002). Since I^2 and H^2 are just monotonic transformation of τ^2 , the confidence intervals for I^2 and H^2 are also exact.

Value

Either a single data frame or a list with two data frames (named `fixed` and `random`) with the following elements:

<code>estimate</code>	estimate of the model coefficient or variance component.
<code>ci.lb</code>	lower bound of the confidence interval.
<code>ci.ub</code>	upper bound of the confidence interval.

For fixed-effects models, the variance component estimates are NA, but the confidence interval bounds are still provided.

Note

The iterative algorithm used as part of the Q-profile 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 upper bound of the interval searched is set to 50 (which should be large enough for most cases). The desired accuracy (`tol`), maximum number of iterations (`maxiter`), and upper bound (`tau2.max`) can be adjusted with `control=list(tol=value, maxiter=value, tau2.max=value)`. One can also adjust the lower bound of the interval searched with `control=list(tau2.min=value)` (the default is to take that value from the object, which is 0 by default). You should only play around with this latter value if you know what you are doing.

It is possible that the lower and upper confidence interval bounds both fall below zero (or whatever value was chosen for `tau2.min`). Since both values then fall outside of the parameter space, the confidence interval then just consists of the null set.

Usually, the estimate of τ^2 from the random/mixed-effects model will fall within the confidence interval 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 confidence interval, it can happen in rare cases that the confidence interval does not actually contain the estimate (trying to explain this to reviewers can be tricky). However, using the empirical Bayes estimator of τ^2 when fitting the model guarantees that the estimate of τ^2 falls within the confidence interval.

The Q-profile method is exact under the assumptions of the random- and mixed-effects models (i.e., normally distributed observed and true outcomes and known sampling variances). These assumptions are usually only approximately true, turning the confidence interval for τ^2 also into an approximation.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- 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**, 157–177.
- Higgins, J. P. T. & Thompson, S. G. (2002). Quantifying heterogeneity in a meta-analysis. *Statistics in Medicine*, **21**, 1539–1558.

Mantel, N. & Haenszel, W. (1959). Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the National Cancer Institute*, **22**, 719–748.

Viechtbauer, W. (2007). Confidence intervals for the amount of heterogeneity in meta-analysis. *Statistics in Medicine*, **26**, 37–52.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/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**, 335–371.

See Also

[rma.uni](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

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

### mixed-effects model with absolute latitude in the model
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=ablat,
          data=dat.bcg, measure="RR", method="REML")

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

cumul

Cumulative Meta-Analysis

Description

The function `cumul` is generic. For suitable model objects, it repeatedly fits the model, adding one observation at a time to the model.

Usage

```
cumul(x, ...)
```

Arguments

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

Details

Currently, there are methods for handling objects of class "rma.uni", "rma.mh", and "rma.peto" with the cumul function. Accordingly, the corresponding methods are called [cumul.rma.uni](#), [cumul.rma.mh](#), and [cumul.rma.peto](#). See the documentation for those functions for more details.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Chalmers, T. C. & Lau, J. (1993). Meta-analytic stimulus for changes in clinical trials. *Statistical Methods in Medical Research*, **2**, 161–172.
- 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**, 45–57.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[cumul.rma.uni](#), [cumul.rma.mh](#), [cumul.rma.peto](#)

cumul.rma.mh

Cumulative Meta-Analysis for rma.mh and rma.peto Objects

Description

The functions `leave1out.rma.mh` and `leave1out.rma.peto` repeatedly fit the specified model, adding one observation (i.e., 2x2 table) at a time to the model.

Usage

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

Arguments

- | | |
|--------|---|
| x | an object of class "rma.mh" or "rma.peto". |
| order | an optional vector with indices giving the desired order for the cumulative meta-analysis. |
| digits | an integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object). |

transf	logical indicating whether odds ratios or relative risks (and the corresponding confidence interval bounds) should be given in the transformed (meaning: raw) units or in terms of log units (the default).
...	other arguments.

Value

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

estimate	estimated coefficients of the model.
se	standard errors of the coefficients. NA if <code>transf=TRUE</code> .
zval	test statistics of the coefficients.
pval	p-values for the test statistics.
ci.lb	lower bounds of the confidence intervals for the coefficients.
ci.ub	upper bounds of the confidence intervals for the coefficients.
QE	test statistics for the tests of heterogeneity.
QEp	p-values for the tests of heterogeneity.

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`.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Chalmers, T. C. & Lau, J. (1993). Meta-analytic stimulus for changes in clinical trials. *Statistical Methods in Medical Research*, **2**, 161–172.
- 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**, 45–57.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[cumul](#), [forest.cumul.rma](#)

Examples

```

### load BCG vaccine data
data(dat.bcg)

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

cumul(res, order=order(dat.bcg$year))
cumul(res, order=order(dat.bcg$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=order(dat.bcg$year))
cumul(res, order=order(dat.bcg$year), transf=TRUE)

```

cumul.rma.uni

*Cumulative Meta-Analysis for rma.uni Objects***Description**

The function `cumul.rma.uni` repeatedly fits the specified model, adding one observation at a time to the model.

Usage

```

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

```

Arguments

<code>x</code>	an object of class "rma.uni".
<code>order</code>	an optional vector with indices giving the desired order for the cumulative meta-analysis.
<code>digits</code>	an integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
<code>transf</code>	an optional argument specifying the name of a function that should be used to transform the model coefficients and interval bounds (e.g., <code>transf=exp</code>). Defaults to <code>FALSE</code> , which means that no transformation is used.
<code>targs</code>	optional arguments needed by the function specified under <code>transf</code> .
<code>...</code>	other arguments.

Details

The model specified by `x` must be a model without moderators (i.e., either a fixed- or a random-effects model).

Value

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

<code>estimate</code>	estimated coefficients of the model.
<code>se</code>	standard errors of the coefficients. NA if <code>transf</code> is used to transform the coefficients.
<code>zval</code>	test statistics of the coefficients.
<code>pval</code>	p-values for the test statistics.
<code>ci.lb</code>	lower bounds of the confidence intervals for the coefficients.
<code>ci.ub</code>	upper bounds of the confidence intervals for the coefficients.
<code>QE</code>	test statistics for the tests of heterogeneity.
<code>QEp</code>	p-values for the tests of heterogeneity.
<code>tau2</code>	estimated amounts of (residual) heterogeneity (only for random-effects models).
<code>I2</code>	values of I^2 (only for random-effects models).
<code>H2</code>	values of H^2 (only for random-effects models).

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`.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Chalmers, T. C. & Lau, J. (1993). Meta-analytic stimulus for changes in clinical trials. *Statistical Methods in Medical Research*, **2**, 161–172.
- 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**, 45–57.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[cumul](#), [forest.cumul.rma](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### calculate log relative risks and corresponding sampling variances
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg,
```

```

data=dat.bcg, append=TRUE)

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

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

```

dat.bcg	<i>Studies on the Effectiveness of the BCG Vaccine for Preventing Tuberculosis</i>
---------	--

Description

Results from 13 clinical trials examining the effectiveness of the Bacillus Calmette-Guerin (BCG) vaccine for preventing tuberculosis.

Usage

```
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 group
cneg	numeric	number of TB negative cases in the control 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 2x2 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 'Refer-

ences’).

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**, 698–702.

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**, 395–411.

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

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

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

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

### mixed-effects model with absolute latitude and publication year as moderators
res <- rma(yi, vi, mods=cbind(ablat, year), data=dat, method="REML")
res

### predicted average relative risks 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$ablat, dat$year)
cor(dat$ablat, dat$year)
```

dat.co2

*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.co2

Format

The data frame contains the following columns:

id	numeric	observation number
paper	numeric	paper number
units	numeric	paper number
genus	character	genus name
species	character	species name
function	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 below)
method	character	CO2 exposure facility (see below)
stock	character	planting stock code
xtrt	character	interacting treatment code (see below)
level	character	interacting treatment level codes (see below)
m1i	numeric	mean plant mass under elevated CO2 level (treatment group)
sd1i	numeric	standard deviation of plant mass under elevated 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**, 1150–1156. (data obtained from *Ecological Archives*, E080-008-S1, at: <http://www.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**, 299–313.

Examples

```
### load CO2 data
data(dat.co2)

### calculate log ratio of means and corresponding sampling variances
dat <- escalc(mli=mli, sd1i=sd1i, n1i=n1i, m2i=m2i, sd2i=sd2i, n2i=n2i,
             measure="ROM", data=dat.co2, append=TRUE)
dat

### meta-analysis of log ratio of means 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.empint

Studies on the Validity of Employment Interviews

Description

Results from 160 studies on the correlation between employment interview assessments and job performance.

Usage

```
dat.empint
```

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 contained in this dataset were obtained from Table A.2 in Rothstein, Sutton, & Borenstein (2005, p. 325-329). Note that the type and struct variables contain some NAs.

Source

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**, 599–616.

Rothstein, H. R., Sutton, A. J., & Borenstein, M. (Eds.). (2005). *Publication bias in meta-analysis: Prevention, assessment, and adjustments*. Chichester, England: Wiley.

Examples

```
### load empint data
data(dat.empint)

### calculate r-to-z transformed correlations and corresponding sampling variances
dat <- escalc(ri=ri, ni=ni, measure="ZCOR", data=dat.empint, append=TRUE)
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(ri=ri, ni=ni, measure="COR", data=dat.empint)
res
```

dat.los

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.los

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**, 321–359.

Examples

```
### load los data
data(dat.los)

### calculate mean differences and corresponding sampling variances
dat <- escalc(m1i=m1i, sd1i=sd1i, n1i=n1i, m2i=m2i, sd2i=sd2i, n2i=n2i,
             measure="MD", data=dat.los, append=TRUE)
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(m1i=m1i, sd1i=sd1i, n1i=n1i, m2i=m2i, sd2i=sd2i, n2i=n2i,
          measure="SMD", data=dat.los, slab=source)
res

### draw forest plot
forest(res, xlim=c(-7,5), alim=c(-3,1))
text(-7, 11, "Study/Source", pos=4)
text( 5, 11, "Observed SMD [95% CI]", pos=2)
```

 dat.warfarin

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.warfarin
```

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 (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 (years) in the placebo/control group
compgrp	character	type of comparison group (placebo or control)
prevtype	character	type of prevention (primary or secondary)
trlnr	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**, 492–501.

Examples

```
### load warfarin data
data(dat.warfarin)

### calculate log incidence rate ratios and corresponding sampling variances
```

```

dat <- escalc(x1i=x1i, x2i=x2i, t1i=t1i, t2i=t2i, measure="IRR",
             data=dat.warfarin, append=TRUE)
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
forest(res, xlim=c(-14, 6), at=log(c(.05, .25, 1, 4)), atransf=exp,
       slab=paste(dat$study, " (", dat$year, ")", sep=""),
       ilab=cbind(paste(dat$x1i, "/", dat$t1i, sep=" "),
                  paste(dat$x2i, "/", dat$t2i, sep=" ")),
       ilab.xpos=c(-8,-5), cex=.85)
op <- par(cex=.85, font=2)
text(-14, 7.5, "Study (Year)", pos=4)
text(6, 7.5, "IRR [95% CI]", pos=2)
text(c(-8,-5), 8.0, c("Strokes /", "Strokes /"))
text(c(-8,-5), 7.5, c("Person-Time", "Person-Time"))
text(c(-8,-5), 8.5, c("Warfarin", "Control"))
segments(x0=-9, y0=8.25, x1=-4, y1=8.25)
par(op)

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

```

escalc

Calculate Effect Size and Outcome Measures

Description

The function can be used to calculate various effect size or outcome measures (and the corresponding sampling variances) that are commonly used in meta-analyses.

Usage

```

escalc(measure, formula, ...)

## Default S3 method:
escalc(measure, formula, ai, bi, ci, di, n1i, n2i,
       x1i, x2i, t1i, t2i, m1i, m2i, sd1i, sd2i,
       xi, mi, ri, ni, ti, data,
       add=1/2, to="only0", vtype="LS", append=FALSE, ...)

## S3 method for class 'formula'

```

```
escalc(measure, formula, weights, data,
       add=1/2, to="only0", vtype="LS", ...)
```

Arguments

measure	a character string indicating which effect size or outcome measure should be calculated. See ‘Details’ for possible options and how the data should then be specified.
formula	when using the formula interface of the function (see ‘Details’ below), a model formula specifying the data structure should be specified via this argument. When not using the formula interface, this argument can be ignored and the data required to calculate the effect sizes or outcomes are then passed to the function via the following set of arguments. See ‘Details’.
weights	vector of weights to specify the group sizes or cell frequencies (only needed when using the formula interface). See ‘Details’.
ai	vector to specify the 2x2 table frequencies (upper left cell). See ‘Details’.
bi	vector to specify the 2x2 table frequencies (upper right cell). See ‘Details’.
ci	vector to specify the 2x2 table frequencies (lower left cell). See ‘Details’.
di	vector to specify the 2x2 table frequencies (lower right cell). See ‘Details’.
n1i	vector to specify the group sizes or row totals (first group/row). See ‘Details’.
n2i	vector to specify the group sizes or row totals (second group/row). See ‘Details’.
x1i	vector to specify the number of cases (first group). See ‘Details’.
x2i	vector to specify the number of cases (second group). See ‘Details’.
t1i	vector to specify the total person-times (first group). See ‘Details’.
t2i	vector to specify the total person-times (second group). See ‘Details’.
m1i	vector to specify the means (first group). See ‘Details’.
m2i	vector to specify the means (second group). See ‘Details’.
sd1i	vector to specify the standard deviations (first group). See ‘Details’.
sd2i	vector to specify the standard deviations (second group). See ‘Details’.
xi	vector to specify the frequencies of the event of interest. See ‘Details’.
mi	vector to specify the frequencies of the complement of the event. See ‘Details’.
ri	vector to specify the raw correlation coefficients. See ‘Details’.
ni	vector to specify the sample sizes. See ‘Details’.
ti	vector to specify the total person-times. See ‘Details’.
data	an optional data frame containing the variables given to the arguments above.
add	a non-negative number indicating the amount to add to zero cells, counts, or frequencies. See ‘Details’.
to	a string indicating when the values under add should be added (either "all", "only0", "if0all", or "none"). See ‘Details’.
vtype	a string indicating the type of sampling variances to calculate (either "LS" or "UB"). See ‘Details’.

append	logical indicating whether the data frame specified via the data argument (if one has been specified) should be returned together with the effect sizes and sampling variances (default is FALSE).
...	other arguments.

Details

There are two interfaces to using the `escalc` function, the default and a formula interface. The two interfaces are described below.

Default Interface:

The default interface works as follows. The argument `measure` is a character string specifying which outcome measure should be calculated (see below for the various options), arguments `ai` through `ni` are then used to specify the needed information to calculate the various measures (depending on the outcome measure, different arguments need to be supplied), and `data` can be used to specify a data frame containing the variables given to the previous arguments. The `add` and `to` arguments may be needed when dealing with 2x2 table data that contain cells with zeros. Finally, the `vtype` argument is used to specify how to calculate the sampling variance estimate (see below).

Effect Size and Outcome Measures for 2x2 Table Data:

Meta-analyses in the health/medical sciences are often based on studies providing data in terms of 2x2 tables. In particular, assume that we have k tables of the form:

	outcome 1	outcome 2	total
group 1	a_i	b_i	n_{1i}
group 2	c_i	d_i	n_{2i}

where a_i , b_i , c_i , and d_i denote the cell frequencies and n_{1i} and n_{2i} 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. The 2x2 table may also be the result of cross-sectional (i.e., multinomial) sampling, so that none of the table margins (except the total sample size $n_{1i}+n_{2i}$) are fixed through the study design.

Depending on the type of design (sampling method), a meta-analysis of 2x2 table data can be based on one of several different outcome measures, including the odds ratio, the relative risk (also called risk ratio), the risk difference, and the arcsine transformed risk difference (for example, for case-control, the odds ratio is the measure of choice, while for RCTs and cohort studies, all of these measures may be applicable). The phi coefficient, Yule's Q , and Yule's Y are additional measures of association for 2x2 table data (although they are not frequently used in meta-analyses). For these outcome measures, one needs to specify either a_i , b_i , c_i , and d_i or alternatively a_i , c_i , n_{1i} , and n_{2i} . The options for the `measure` argument are then:

- "RR": The *log relative risk* is equal to the log of $(a_i/n_{1i})/(c_i/n_{2i})$.
- "OR": The *log odds ratio* is equal to the log of $(a_i*d_i)/(b_i*c_i)$.
- "RD": The *risk difference* is equal to $(a_i/n_{1i})-(c_i/n_{2i})$.
- "AS": The arcsine transformation is a variance stabilizing transformation for proportions.

The *arcsine transformed risk difference* is equal to $\text{asin}(\sqrt{a_i/n_{1i}}) - \text{asin}(\sqrt{c_i/n_{2i}})$. See Ruecker et al. (2009) for a discussion of this and other outcome measures for 2x2 table data.

- "PETO": The *log odds ratio estimated with Peto's method* (see Yusuf et al., 1985) is equal to $(a_i - s_i * n_{1i} / n_i) / ((s_i * t_i * n_{1i} * n_{2i}) / (n_i^2 * (n_i - 1)))$, where $s_i = a_i + c_i$, $t_i = b_i + d_i$, and $n_i = n_{1i} + n_{2i}$.
- "PHI": The *phi coefficient* is equal to $(a_i * d_i - b_i * c_i) / \sqrt{n_{1i} * n_{2i} * s_i * t_i}$, where $s_i = a_i + c_i$ and $t_i = b_i + d_i$.
- "YUQ": *Yule's Q* is equal to $(o_i - 1) / (o_i + 1)$, where o_i is the odds ratio.
- "YUY": *Yule's Y* is equal to $(\sqrt{o_i} - 1) / (\sqrt{o_i} + 1)$, where o_i is the odds ratio.

Note that the log is taken of the relative risk and the odds ratio, which makes these outcome measures symmetric around 0 and helps to make the distribution of these outcome measure closer to normal.

Cell entries with a zero can be problematic, especially for the relative risk and the odds ratio. Adding a small constant to the cells of the 2x2 tables is a common solution to this problem. When `to="all"`, the value of `add` is added to each cell of all 2x2 tables. When `to="only0"` (the default), the value of `add` (the default is 1/2) is added to each cell of the 2x2 tables with at least one cell equal to 0. When `to="if0all"`, the value of `add` is added to each cell of all 2x2 tables, but only when there is at least one 2x2 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 `Inf` value is recoded to `NA`).

An example dataset corresponding to data of this type is provided in [dat.bcg](#).

Incidence Rate Ratios and Differences:

Epidemiological studies often compare the incidence rates (i.e., the rate of occurrence of a particular outcome, e.g., a certain disease, over a particular period of time) of two different groups (e.g., exposed, not exposed). In particular, assume that we have k tables of the form:

	cases	person-time
group 1	x_{1i}	t_{1i}
group 2	x_{2i}	t_{2i}

where x_{1i} and x_{2i} denote the number of cases in the first and the second group, respectively, and t_{1i} and t_{2i} the corresponding total person-times at risk. Commonly used effect size or outcome measures in this context are the ratio or the difference between the two incidence rates. The options for the measure argument are then:

- "IRR": The *log incidence rate ratio* is equal to the log of $(x_{1i}/t_{1i}) / (x_{2i}/t_{2i})$.
- "IRD": The *incidence rate difference* is equal to $(x_{1i}/t_{1i}) - (x_{2i}/t_{2i})$.
- "IRSD": The square-root transformation is a variance stabilizing transformation for incidence rates. The *square-root transformed incidence rate difference* is equal to $\sqrt{x_{1i}/t_{1i}} - \sqrt{x_{2i}/t_{2i}}$.

Note that the log is taken of the incidence rate ratio, which makes this outcome measure symmetric around 0 and helps to make its distribution closer to normal.

Studies with zero cases in one or both groups can be problematic, especially for the incidence rate ratio. Adding a small constant to the number of cases is a common solution to this problem. When `to="all"`, the value of `add` is added to x_{1i} and x_{2i} in all k studies. When `to="only0"` (the default), the value of `add` (the default is 1/2) is added to x_{1i} and x_{2i} only in the studies

that have zero cases in one or both groups. When `to="if0a11"`, the value of `add` is added to x_{1i} and x_{2i} in all k studies, but only when there is at least one study with zero cases in one or both groups. Setting `to="none"` or `add=0` has the same effect: No adjustment to the observed number of cases 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 `Inf` value is recoded to `NA`).

An example dataset corresponding to data of this type is provided in [dat.warfarin](#).

Mean Differences, Standardized Mean Differences, and Ratio of Means:

The raw mean difference, the standardized mean difference, and the ratio of means (also called response ratio) are useful effect size measures when meta-analyzing a set of studies comparing two experimental groups (e.g., treatment and control groups) or two naturally occurring groups (e.g., men and women) with respect to some quantitative (and ideally normally distributed) dependent variable. For these outcome measures, m_{1i} and m_{2i} are used to specify the means of the two groups, sd_{1i} and sd_{2i} the standard deviations of the scores in the two groups, and n_{1i} and n_{2i} the sample sizes of the two groups. The options for the measure argument are then:

- "MD": The *raw mean difference* is equal to $m_{1i} - m_{2i}$.
- "SMD": The *standardized mean difference* is equal to $(m_{1i} - m_{2i}) / spi$, where spi is the pooled standard deviation of the two groups (which is calculated inside of the function based on sd_{1i} and sd_{2i}). The standardized mean difference is automatically corrected for its slight positive bias within the function (see Hedges & Olkin, 1985). When `vtype="LS"`, the sampling variances are calculated based on the large sample approximation. Alternatively, the unbiased estimates of the sampling variances can be obtained with `vtype="UB"`.
- "ROM": The *log transformed ratio of means* is equal to $\log(m_{1i}/m_{2i})$. See Hedges et al. (1999) for more details on this outcome measure.

Note that the log is taken of the ratio of means, which makes this outcome measures symmetric around 0 and helps to make the distribution of this measure closer to normal (however, if m_{1i} and m_{2i} have opposite signs, this outcome measure cannot be computed).

An example dataset corresponding to data of this type is provided in [dat.los](#) (for mean differences and standardized mean differences). An example dataset showing the use of the ratio of means measure is provided in [dat.co2](#).

Raw and Transformed Correlation Coefficients:

Another frequently used outcome measure in meta-analyses is the correlation coefficient, which is used to measure the strength of the (linear) relationship between two quantitative variables. Here, one needs to specify r_i , the vector with the raw correlation coefficients, and n_i , the corresponding sample sizes. The options for the measure argument are then:

- "COR": The *raw correlation coefficient* is simply equal to r_i as supplied to the function. When `vtype="LS"`, the sampling variances are calculated based on the large sample approximation. Alternatively, an approximation to the unbiased estimates of the sampling variances can be obtained with `vtype="UB"` (see Hedges, 1989).
- "UCOR": The *unbiased estimate of the correlation coefficient* is obtained by correcting the raw correlation coefficient for its slight negative bias (based on equation 2.7 in Olkin & Pratt, 1958). Again, `vtype="LS"` and `vtype="UB"` can be used to choose between the large sample approximation or approximately unbiased estimates of the sampling variances.
- "ZCOR": Fisher's *r-to-z transformation* is a variance stabilizing transformation for correlation coefficients with the added benefit of also being a rather effective normalizing transformation (Fisher, 1921). The *Fisher's r-to-z transformed correlation coefficient* is equal to $1/2 * \log((1+r_i)/(1-r_i))$.

An example dataset corresponding to data of this type is provided in [dat.empint](#).

Proportions and Transformations Thereof:

When the studies provide data for single groups with respect to a dichotomous dependent variable, then the raw proportion, the logit transformed proportion, the arcsine transformed proportion, and the Freeman-Tukey (double arcsine) transformed proportion are useful outcome measures (the log transformed proportion is also a possibility, but not frequently used in meta-analyses). Here, one needs to specify x_i and n_i , denoting the number of individuals experiencing the event of interest and the total number of individuals, respectively. Instead of specifying n_i , one can use m_i to specify the number of individuals that do not experience the event of interest. The options for the measure argument are then:

- "PR": The *raw proportion* is equal to x_i/n_i .
- "PLN": The *log transformed proportion* is equal to the log of x_i/n_i .
- "PLO": The *logit transformed proportion* is equal to the log of $x_i/(n_i-x_i)$ (i.e., the log of the odds).
- "PAS": The arcsine transformation is a variance stabilizing transformation for proportions. The *arcsine transformed proportion* is equal to $\text{asin}(\sqrt{x_i/n_i})$.
- "PFT": Another variance stabilizing transformation for proportions was suggested by Freeman & Tukey (1950). The *Freeman-Tukey double arcsine transformed proportion* is equal to $1/2 * (\text{asin}(\sqrt{x_i/(n_i+1)}) + \text{asin}(\sqrt{(x_i+1)/(n_i+1)}))$.

Zero cell entries can be problematic for certain outcome measures. When `to="all"`, the value of `add` is added to x_i and m_i in all k studies. When `to="only0"` (the default), the value of `add` (the default is 1/2) is added to x_i and m_i only for studies where x_i or m_i is equal to 0. When `to="if0all"`, the value of `add` is added in all k studies, but only when there is at least one study with a zero value for x_i or m_i . Setting `to="none"` or `add=0` again means that no adjustment to the observed values is made.

Incidence Rates and Transformations Thereof:

Instead of proportions, we may also be interested in aggregating individual incidence rates. Here, one needs to specify x_i and t_i , denoting the number of individuals experiencing the event of interest and the total person-time at risk, respectively. The options for the measure argument are then:

- "IR": The *raw incidence rate* is equal to x_i/t_i .
- "IRLN": The *log transformed incidence rate* is equal to the log of x_i/t_i .
- "IRS": The square-root transformation is a variance stabilizing transformation for incidence rates. The *square-root transformed incidence rate* is equal to $\sqrt{x_i/t_i}$.
- "IRFT": Another variance stabilizing transformation for incidence rates can be based on Freeman & Tukey (1950). The *Freeman-Tukey transformed incidence rate* is equal to $\sqrt{x_i/t_i} + \sqrt{x_i/t_i + 1/t_i}$.

Studies with zero cases can be problematic, especially for the log transformed incidence rate. Adding a small constant to the number of cases is a common solution to this problem. When `to="all"`, the value of `add` is added to x_i in all k studies. When `to="only0"` (the default), the value of `add` (the default is 1/2) is added to x_i only in the studies that have zero cases. When `to="if0all"`, the value of `add` is added to x_i in all k studies, but only when there is at least one study with zero cases. Setting `to="none"` or `add=0` has the same effect: No adjustment to the observed number of cases 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 Inf value is recoded to NA).

Formula Interface:

The formula interface works as follows. As above, the argument `measure` is a character string specifying which outcome measure should be calculated. The `formula` argument is then used to specify the data structure as a multipart formula. The `data` argument can be used to specify a data frame containing the variables in the formula. The `add`, `to`, and `vtype` arguments work as described above.

Effect Size and Outcome Measures for 2x2 Table Data:

For 2x2 table data, the formula argument takes the form `outcome ~ group | study`, where `group` is a two-level factor specifying the rows of the tables, `outcome` is a two-level factor specifying the columns of the tables (the two possible outcomes), and `study` is a factor specifying the study factor. The `weights` argument is used to specify the frequencies in the various cells.

Incidence Rate Ratios and Differences:

For these outcome measures, the formula argument takes the form `cases/times ~ group | study`, where `group` is a two-level factor specifying the group factor and `study` is a factor specifying the study factor. The left-hand side of the formula is composed of two parts, with the first variable for the number of cases and the second variable for the person-time at risk.

Mean Differences, Standardized Mean Differences, and Ratio of Means:

For these outcome measures, the formula argument takes the form `means/sds ~ group | study`, where `group` is a two-level factor specifying the group factor and `study` is a factor specifying the study factor. The left-hand side of the formula is composed of two parts, with the first variable for the means and the second variable for the standard deviations. The `weights` argument is used to specify the sample sizes in the groups.

Raw and Transformed Correlation Coefficients:

For these outcome measures, the formula argument takes the form `outcome ~ 1 | study`, where `outcome` is used to specify the observed correlations and `study` is a factor specifying the study factor. The `weights` argument is used to specify the sample sizes.

Proportions and Transformations Thereof:

For these outcome measures, the formula argument takes the form `outcome ~ 1 | study`, where `outcome` is a two-level factor specifying the columns of the tables (the two possible outcomes) and `study` is a factor specifying the study factor. The `weights` argument is used to specify the frequencies in the various cells.

Incidence Rates and Transformations Thereof:

For these outcome measures, the formula argument takes the form `cases/times ~ 1 | study`, where `study` is a factor specifying the study factor. The left-hand side of the formula is composed of two parts, with the first variable for the number of cases and the second variable for the person-time at risk.

Value

A data frame with the following elements:

<code>yi</code>	value of the effect size or outcome measure.
<code>vi</code>	corresponding (estimated) sampling variance.

If `append=TRUE` and a data frame was specified via the `data` argument, then `yi` and `vi` are appended to this data frame.

Note

For standard meta-analyses using the typical (wide-format) data layout (i.e., one row in the dataset per study), the default interface is typically easier to use. The advantage of the formula interface is that it can, in principle, handle more complicated data structures (e.g., studies with more than two treatment groups or more than two outcomes). While such functionality is currently not implemented, this may be the case in the future.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

- Cooper, H. C., Hedges, L. V., & Valentine, J. C. (Eds.) (2009). *The handbook of research synthesis and meta-analysis* (2nd ed.). New York: Russell Sage Foundation.
- Fisher, R. A. (1921). On the “probable error” of a coefficient of correlation deduced from a small sample. *Metron*, **1**, 1–32.
- Freeman, M. F. & Tukey, J. W. (1950). Transformations related to the angular and the square root. *Annals of Mathematical Statistics*, **21**, 607–611.
- Hedges, L. V. (1989). An unbiased correction for sampling error in validity generalization studies. *Journal of Applied Psychology*, **74**, 469–477.
- Hedges, L. V. & Olkin, I. (1985). *Statistical methods for meta-analysis*. San Diego, CA: Academic Press.
- Hedges, L. V., Gurevitch, J., & Curtis, P. S. (1999). The meta-analysis of response ratios in experimental ecology. *Ecology*, **80**, 1150–1156.
- Ruecker, 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**, 721–738.
- Olkin, I. & Pratt, J. W. (1958). Unbiased estimation of certain correlation coefficients. *Annals of Mathematical Statistics*, **29**, 201–211.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/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**, 335–371.

See Also

[rma.uni](#), [rma.mh](#), [rma.peto](#)

Examples

```

### load BCG vaccine data
data(dat.bcg)

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

### using formula interface (first rearrange data into required format)
k <- length(dat.bcg$trial)
dat.fm <- data.frame(study=factor(rep(1:k, each=4)))
dat.fm$grp <- factor(rep(c("T","T","C","C"), k), levels=c("T","C"))
dat.fm$out <- factor(rep(c("+","-","+","-"), k), levels=c("+","-"))
dat.fm$freq <- with(dat.bcg, c(rbind(tpos, tneg, cpos, cneg)))
dat.fm
escalc(out ~ grp | study, weights=freq, data=dat.fm, measure="RR")

```

fitstats

*Fit Statistics and Information Criteria***Description**

The function `fitstats` is generic. It extracts the log likelihood, deviance, and information criteria (e.g., AIC and BIC) for suitable objects.

Usage

```
fitstats(object, ...)
```

Arguments

<code>object</code>	an object for which fit statistics and information criteria can be calculated. See ‘Details’.
<code>...</code>	other arguments.

Details

Currently, there is only a method for handling objects of class `"rma"` with the `fitstats` function. Accordingly, the corresponding method is called `fitstats.rma`. See the documentation for that function for more details.

Value

Log likelihood, deviance, AIC, and BIC values either under the regular or restricted likelihood.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[fitstats.rma](#)

 fitstats.rma

Fit Statistics and Information Criteria for rma Objects

Description

Functions to extract the log likelihood, deviance, AIC, and BIC values from objects of class "rma".

Usage

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

Arguments

object	an object of class "rma".
REML	logical indicating 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 specifying the penalty per parameter to be used. The default k=2 is the classical AIC. See AIC for more details.
...	other arguments.

Value

For `fitstats.rma`, a column vector with the (restricted) log likelihood, deviance, AIC, and BIC values.

For `logLik.rma`, an object of class "logLik", providing the (restricted) log likelihood of the model evaluated at the estimated coefficient(s).

For `deviance.rma`, a numeric value with the corresponding deviance.

For `AIC.rma`, a numeric value with the corresponding AIC.

For `BIC.rma`, a numeric value with the corresponding BIC.

Note

When a random/mixed-effects model is fitted, τ^2 is counted as an additional parameter in the calculation of the AIC and BIC. Also, the fixed effects are counted as parameters in the calculation of the AIC and BIC even when using REML estimation.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.uni](#), [rma.mh](#), [rma.peto](#), [anova.rma.uni](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a random-effects model
res1 <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg,
           data=dat.bcg, measure="RR", method="ML")

### mixed-effects model with two moderators (latitude and publication year)
res2 <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat, year),
           data=dat.bcg, measure="RR", method="ML")

fitstats(res1)
fitstats(res2)

logLik(res1)
logLik(res2)
```

```
deviance(res1)  
deviance(res2)
```

```
AIC(res1)  
AIC(res2)
```

```
BIC(res1)  
BIC(res2)
```

fitted.rma

Fitted Values for rma Objects

Description

The function calculates 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>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[predict.rma](#), [blup.rma.uni](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a mixed-effects model
### with two moderators (absolute latitude and publication year)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat, year),
          data=dat.bcg, measure="RR", method="REML")
fitted(res)
```

forest

Forest Plots

Description

The function `forest` is generic. It can be used to create forest plots.

Usage

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

Arguments

`x` either an object of class `"rma"`, a vector with the observed effect size 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 size or outcomes. The corresponding method is then `forest.default`.

Finally, object `x` could 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>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Lewis, S. & Clarke, M. (2001). Forest plots: Trying to see the wood and the trees. *British Medical Journal*, **322**, 1479–1480.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[forest.rma](#), [forest.default](#), [forest.cumul.rma](#)

forest.cumul.rma

Forest Plots 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, xlim=NULL, alim=NULL,
       ylim=NULL, at=NULL, steps=5, level=x$level, digits=2, refline=0,
       xlab=NULL, ilab=NULL, ilab.xpos=NULL, ilab.pos=NULL, transf=FALSE,
       atranf=FALSE, targs, rows=NULL, efac=1, pch=15, psize=1,
       cex=NULL, cex.lab=NULL, cex.axis=NULL, ...)
```

Arguments

x	an object of class "cumul.rma".
annotate	logical specifying whether annotations should be added to the plot (default is TRUE).
xlim	horizontal limits of the plot region. Defaults to NULL, which means that the function tries to set the horizontal plot limits to some sensible values.
alim	the actual x axis limits. Defaults to NULL, which means that the function tries to set the x axis limits to some sensible values.
ylim	the y limits of the plot. Defaults to NULL, which means that the function tries to set the y axis limits to some sensible values.
at	position of the x axis tick marks and corresponding labels. Defaults to NULL, which means that the function tries to set the tick mark positions/labels to some sensible values.
steps	the number of tick marks for the x axis (default is 5). Ignored when the user specifies the positions via the at argument.
level	numerical value between 0 and 100 specifying the confidence interval level (the default is to take the value from the object).

<code>digits</code>	integer value specifying the number of decimal places to which the tick mark labels of the x axis and the annotations should be rounded (default is 2).
<code>refline</code>	value at which a vertical ‘reference’ line should be drawn (default is 0). The line can be suppressed by setting this argument to NA.
<code>xlab</code>	title for the x axis. If NULL (default), the function tries to set an appropriate axis title.
<code>ilab</code>	an optional vector or matrix of character strings providing additional information about the studies.
<code>ilab.xpos</code>	vector of numerical value(s) specifying the x axis position(s) of the character vector(s) given via <code>ilab</code> (must be specified if <code>ilab</code> is not NULL).
<code>ilab.pos</code>	integer(s) (either 1, 2, 3, or 4) specifying the alignment of the character vector(s) given via <code>ilab</code> (2 means right, 4 mean left aligned). The default is to center the labels.
<code>transf</code>	an optional argument specifying the name of a function that should be used to transform the observed effect sizes, summary estimates, fitted values, and confidence interval bounds (e.g., <code>transf=exp</code>). Defaults to FALSE, which means that no transformation is used.
<code>atransf</code>	an optional argument specifying the name of a function that should be used to transform the x axis labels and annotations (e.g., <code>transf=exp</code>). Defaults to FALSE, which means that no transformation is used.
<code>targs</code>	optional arguments needed by the function specified via <code>transf</code> or <code>atransf</code> .
<code>rows</code>	an optional vector specifying the rows (or more generally, the horizontal positions) for plotting the outcomes. Defaults to NULL, which means that the function sets this value automatically. Can also be a single value specifying the row (horizontal position) of the first outcome (the remaining outcomes are then plotted below this starting row).
<code>efac</code>	vertical expansion factor for arrows, confidence interval limits, and the symbol used to denote summary estimates. The default value of 1 should usually work okay.
<code>pch</code>	plotting symbol to use for the observed effect sizes or outcomes. By default, a filled square is used. See points for other options. Can also be a vector of values.
<code>psize</code>	an optional vector with point sizes for the observed effect sizes or outcomes. Default is 1.
<code>cex</code>	an optional character and symbol expansion factor. If NULL (default), the function tries to set this to a sensible value.
<code>cex.lab</code>	an optional expansion factor for the x axis titel. If NULL (default), the function tries to set this to a sensible value.
<code>cex.axis</code>	an optional expansion factor for the x axis labels. If NULL (default), the function tries to set this to a sensible value.
<code>...</code>	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 tweak these in certain circumstances. In particular, 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 (you will have to call the function again after the stretching, so that the label/symbol sizes are properly adjusted).

If the horizontal plot and/or x axis limits are set by the user, then the horizontal plot limits (`xlim`) must be at least as wide as the x axis limits (`xlim`). Moreover, the x axis limits must encompass the observed effect sizes or outcomes. These restrictions are enforced inside the function.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Chalmers, T. C. & Lau, J. (1993). Meta-analytic stimulus for changes in clinical trials. *Statistical Methods in Medical Research*, **2**, 161–172.
- 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**, 45–57.
- Lewis, S. & Clarke, M. (2001). Forest plots: Trying to see the wood and the trees. *British Medical Journal*, **322**, 1479–1480.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[forest](#), [cumul](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

### random-effects model
res <- rma(yi, vi, data=dat, method="REML", slab=paste(author, year, sep=", "))

x <- cumul(res, order=order(dat$year))
```

```

forest(x)

### meta-analysis of the (log) relative risks using the Mantel-Haenszel method
res <- rma.mh(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg,
             measure="RR", slab=paste(author, year, sep=", "))
x <- cumul(res, order=order(dat$year))
forest(x)

```

forest.default

Forest Plots

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, showweight=FALSE,
       xlim=NULL, alim=NULL, ylim=NULL, at=NULL, steps=5,
       level=95, digits=2, refline=0, xlab=NULL,
       slab=NULL, ilab=NULL, ilab.xpos=NULL, ilab.pos=NULL,
       subset=NULL, transf=FALSE, atranf=FALSE, targs,
       rows=NULL, efac=1, pch=15, psize=NULL,
       cex=NULL, cex.lab=NULL, cex.axis=NULL, ...)

```

Arguments

<code>x</code>	a vector of length k with the observed effect sizes or outcomes.
<code>vi</code>	a vector of length k with the corresponding sampling variances.
<code>sei</code>	a vector of length k with the corresponding standard errors. (note: only one of the two, <code>vi</code> or <code>sei</code> , needs to be specified)
<code>ci.lb</code>	a vector of length k with the corresponding lower confidence intervals bounds. Not needed if <code>vi</code> or <code>sei</code> is specified. See 'Details'.
<code>ci.ub</code>	a vector of length k with the corresponding upper confidence intervals bounds. Not needed if <code>vi</code> or <code>sei</code> is specified. See 'Details'.
<code>annotate</code>	logical specifying whether annotations should be added to the plot (default is TRUE).
<code>showweight</code>	logical specifying whether the annotations should also include inverse variance weights (default is FALSE).
<code>xlim</code>	horizontal limits of the plot region. Defaults to NULL, which means that the function tries to set the horizontal plot limits to some sensible values.
<code>alim</code>	the actual x axis limits. Defaults to NULL, which means that the function tries to set the x axis limits to some sensible values.

<code>ylim</code>	the y limits of the plot. Defaults to NULL, which means that the function tries to set the y axis limits to some sensible values.
<code>at</code>	position of the x axis tick marks and corresponding labels. Defaults to NULL, which means that the function tries to set the tick mark positions/labels to some sensible values.
<code>steps</code>	the number of tick marks for the x axis (default is 5). Ignored when the user specifies the positions via the <code>at</code> argument.
<code>level</code>	numerical value between 0 and 100 specifying the confidence interval level (default is 95).
<code>digits</code>	integer value specifying the number of decimal places to which the tick mark labels of the x axis and the annotations should be rounded (default is 2).
<code>refline</code>	value at which a vertical ‘reference’ line should be drawn (default is 0). The line can be suppressed by setting this argument to NA.
<code>xlab</code>	title for the x axis. If NULL (default), the function tries to set an appropriate axis title.
<code>slab</code>	an optional vector with unique labels for the k studies. Defaults to NULL, which means that simple labels are created within the function. To suppress labels, set this argument to NA.
<code>ilab</code>	an optional vector or matrix of character strings providing additional information about the studies.
<code>ilab.xpos</code>	numerical value(s) specifying the x axis position(s) of the character vector(s) given via <code>ilab</code> (must be specified if <code>ilab</code> is not NULL).
<code>ilab.pos</code>	integer(s) (either 1, 2, 3, or 4) specifying the alignment of the character vector(s) given via <code>ilab</code> (2 means right, 4 mean left aligned). The default is to center the labels.
<code>subset</code>	an optional vector indicating the subset of studies that should be used for the plot. This can be a logical vector of length k or a numeric vector indicating the indices of the observations to include. Note that this argument can also be used for reordering the studies.
<code>transf</code>	an optional argument specifying the name of a function that should be used to transform the observed effect sizes or outcomes and corresponding confidence interval bounds (e.g., <code>transf=exp</code>). Defaults to FALSE, which means that no transformation is used.
<code>atransf</code>	an optional argument specifying the name of a function that should be used to transform the x axis labels and annotations (e.g., <code>transf=exp</code>). Defaults to FALSE, which means that no transformation is used.
<code>targs</code>	optional arguments needed by the function specified via <code>transf</code> or <code>atransf</code> .
<code>rows</code>	an optional vector specifying the rows (or more generally, the horizontal positions) for plotting the outcomes. Defaults to NULL, which means that the function sets this value automatically. Can also be a single value specifying the row (horizontal position) of the first outcome (the remaining outcomes are then plotted below this starting row).
<code>efac</code>	vertical expansion factor for arrows, confidence interval limits, and the symbol used to denote summary estimates. The default value of 1 should usually work okay.

pch	plotting symbol to use for the observed effect sizes or outcomes. By default, a filled square is used. See points for other options. Can also be a vector of values.
psize	an optional vector with point sizes for the observed effect sizes or outcomes. Default is NULL, which means that the point sizes are drawn proportional to the precision of the estimates.
cex	an optional character and symbol expansion factor. If NULL (default), the function tries to set this to a sensible value.
cex.lab	an optional expansion factor for the x axis titel. If NULL (default), the function tries to set this to a sensible value.
cex.axis	an optional expansion factor for the x axis labels. If NULL (default), the function tries to set this to a sensible value.
...	other arguments.

Details

The plot shows the individual observed effect sizes or outcomes with corresponding confidence intervals. To use the function, one should specify the observed effect sizes or 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 effect sizes or outcomes together with the corresponding confidence interval bounds (via the `ci.lb` and `ci.ub` arguments).

With the `transf` argument, the observed effect sizes or outcomes and corresponding confidence interval bounds can be transformed with an arbitrary function. For example, if log odds ratios are provided to the function, 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`). The examples below illustrate the use of these arguments.

Summary estimates can also 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 tweak these in certain circumstances. In particular, if the number of studies is very 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 (you will have to call the function again after the stretching, so that the label/symbol sizes are properly adjusted).

If the horizontal plot and/or x axis limits are set by the user, then the horizontal plot limits (`xlim`) must be at least as wide as the x axis limits (`xlim`). Moreover, the x axis limits must encompass the observed effect sizes or outcomes. These restrictions are enforced inside the function.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

- Lewis, S. & Clarke, M. (2001). Forest plots: Trying to see the wood and the trees. *British Medical Journal*, **322**, 1479–1480.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[forest](#), [addpoly](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

### forest plot of the observed log relative risks
forest(dat$yi, dat$vi)

### forest plot of the observed relative risks
forest(dat$yi, dat$vi, slab=paste(dat$author, dat$year, sep=", "), transf=exp,
      alim=c(0,2), steps=5, xlim=c(-2,3.5), refline=1)

### forest plot of the observed relative risks
forest(dat$yi, dat$vi, slab=paste(dat$author, dat$year, sep=", "), atranf=exp,
      at=log(c(.05, .25, 1, 4, 20)), xlim=c(-10,8))

### see also examples for the forest.rma function
```

forest.rma

Forest Plots 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, addcred=FALSE, showweight=FALSE,
      xlim=NULL, alim=NULL, ylim=NULL, at=NULL, steps=5,
      level=x$level, digits=2, refline=0, xlab=NULL,
      slab=NULL, mlab=NULL, ilab=NULL, ilab.xpos=NULL, ilab.pos=NULL,
      order=NULL, transf=FALSE, atranf=FALSE, targs, rows=NULL,
      efac=1, pch=15, psize=NULL, col="darkgray", border="darkgray",
      cex=NULL, cex.lab=NULL, cex.axis=NULL, ...)
```

Arguments

x	an object of class "rma".
annotate	logical specifying whether annotations should be added to the plot (default is TRUE).
addfit	logical specifying whether the summary estimate (for models without moderators) or fitted values (for models with moderators) should be added to the plot. See 'Details'.
addcred	logical specifying whether the bounds of the credibility interval should be added to the plot. See 'Details'.
showweight	logical specifying whether the annotations should also include the weights given to the observed effects or outcomes during the model fitting (default is FALSE).
xlim	horizontal limits of the plot region. Defaults to NULL, which means that the function tries to set the horizontal plot limits to some sensible values.
alim	the actual x axis limits. Defaults to NULL, which means that the function tries to set the x axis limits to some sensible values.
ylim	the y limits of the plot. Defaults to NULL, which means that the function tries to set the y axis limits to some sensible values.
at	position of the x axis tick marks and corresponding labels. Defaults to NULL, which means that the function tries to set the tick mark positions/labels to some sensible values.
steps	the number of tick marks for the x axis (default is 5). Ignored when the user specifies the positions via the at argument.
level	numerical value between 0 and 100 specifying the confidence interval level (the default is to take the value from the object).
digits	integer value specifying the number of decimal places to which the tick mark labels of the x axis and the annotations should be rounded (default is 2).
refline	value at which a vertical 'reference' line should be drawn (default is 0). The line can be suppressed by setting this argument to NA.
xlab	title for the x axis. If NULL (default), the function tries to set an appropriate axis title.
slab	an optional vector with unique labels for the k studies. Defaults to NULL, which means that 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	an optional character string giving a label to the summary estimate from a fixed- or random-effects model. Defaults to NULL, which means that the label is created within the function.
ilab	an optional vector or matrix of character strings providing additional information about the studies.
ilab.xpos	vector of numerical value(s) specifying the x axis position(s) of the character vector(s) given via ilab (must be specified if ilab is not NULL).

<code>ilab.pos</code>	integer(s) (either 1, 2, 3, or 4) specifying the alignment of the character vector(s) given via <code>ilab</code> (2 means right, 4 mean left aligned). The default is to center the labels.
<code>order</code>	an optional character string specifying how the studies should be ordered. See ‘Details’.
<code>transf</code>	an optional argument specifying the name of a function that should be used to transform the observed effect sizes, summary estimates, fitted values, and confidence interval bounds (e.g., <code>transf=exp</code>). Defaults to <code>FALSE</code> , which means that no transformation is used.
<code>atransf</code>	an optional argument specifying the name of a function that should be used to transform the x axis labels and annotations (e.g., <code>transf=exp</code>). Defaults to <code>FALSE</code> , which means that no transformation is used.
<code>targs</code>	optional arguments needed by the function specified via <code>transf</code> or <code>atransf</code> .
<code>rows</code>	an optional vector specifying the rows (or more generally, the horizontal positions) for plotting the outcomes. Defaults to <code>NULL</code> , which means that the function sets this value automatically. Can also be a single value specifying the row (horizontal position) of the first outcome (the remaining outcomes are then plotted below this starting row).
<code>efac</code>	vertical expansion factor for arrows, confidence interval limits, and the symbol used to denote summary estimates. The default value of 1 should usually work okay.
<code>pch</code>	plotting symbol to use for the observed effect sizes or outcomes. By default, a filled square is used. See points for other options. Can also be a vector of values.
<code>psize</code>	an optional vector with point sizes for the observed effect sizes or outcomes. Default is <code>NULL</code> , which means that the point sizes are drawn proportional to the precision of the estimates.
<code>cex</code>	an optional character and symbol expansion factor. If <code>NULL</code> (default), the function tries to set this to a sensible value.
<code>cex.lab</code>	an optional expansion factor for the x axis titel. If <code>NULL</code> (default), the function tries to set this to a sensible value.
<code>cex.axis</code>	an optional expansion factor for the x axis labels. If <code>NULL</code> (default), the function tries to set this to a sensible value.
<code>col</code>	character string specifying the name of a color to use for the fitted values (“darkgray” by default).
<code>border</code>	character string specifying the name of a color to use for the border of the fitted values (“darkgray” by default).
<code>...</code>	other arguments.

Details

The plot shows the individual observed effect sizes or outcomes with corresponding confidence intervals.

For fixed- and random-effects models (i.e., for models without moderators), a polygon is added to the bottom of the forest plot, showing the summary estimate based on the model (with the outer

edges of the polygon indicating the confidence interval limits). For random-effects models and if `addcred=TRUE`, a dashed line indicates the (approximate) bounds of the credibility interval (where 95% of the true effects are estimated to fall).

For models involving moderators, the fitted value for each study is added as a polygon to the plot. With the `col` and `border` arguments, one can change the (border) color of these polygons.

With the `transf` argument, the observed effect sizes, summary estimate, fitted values, and interval bounds can be transformed with an arbitrary function. For example, when analyzing 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`). The examples below illustrate the use of these arguments.

The studies can be reordered with the `order` argument:

- `order="obs"`: the studies are ordered by the observed effect sizes,
- `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 vector with indices specifying the desired order (see examples below).

Additional summary estimates can also 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 tweak these in certain circumstances. In particular, 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 (you will have to call the function again after the stretching, so that the label/symbol sizes are properly adjusted).

If the horizontal plot and/or x axis limits are set by the user, then the horizontal plot limits (`xlim`) must be at least as wide as the x axis limits (`xlim`). Moreover, the x axis limits must encompass the observed effect sizes or outcomes. These restrictions are enforced inside the function.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Lewis, S. & Clarke, M. (2001). Forest plots: Trying to see the wood and the trees. *British Medical Journal*, **322**, 1479–1480.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[forest](#), [addpoly](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a random-effects model
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, measure="RR",
          slab=paste(author, year, sep=", "), method="REML")

### several forest plots illustrating the use of various arguments
forest(res)
forest(res, order=order(dat.bcg$ablat))
forest(res, transf=exp, alim=c(0,6), steps=4, xlim=c(-8,12), refline=1)
forest(res, atranf=exp, at=log(c(.05,.25,1,4,20)), xlim=c(-8,8),
      order="prec", showweight=TRUE, cex=.8)

### forest plot with extra annotations
forest(res, slab=paste(dat.bcg$author, dat.bcg$year, sep=", "),
      xlim=c(-16, 6), at=log(c(.05, .25, 1, 4)), atranf=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)
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"))
text(-16, 15, "Author(s) and Year", pos=4)
text(6, 15, "Relative Risk [95% CI]", pos=2)
par(op)

### mixed-effects model with absolute latitude in the model
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=ablat, data=dat.bcg,
          measure="RR", method="REML", 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, atranf=exp, at=log(c(.05,.25,1,4)))
op <- par(font=2)
text(-9, 15, "Author(s) and Year", pos=4, cex=.8)
text( 5, 15, "Observed RR [95% CI]", pos=2, cex=.8)
text(-4, 15, "Latitude", cex=.8)
par(op)
```

```

### forest plot with subgrouping of studies
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, measure="RR",
          slab=paste(author, year, sep=", "), method="REML")
forest(res, slab=paste(dat.bcg$author, dat.bcg$year, sep=", "),
       xlim=c(-16, 6), at=log(c(.05, .25, 1, 4)), atranf=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=order(dat.bcg$alloc), rows=c(1:2,5:11,14:17))
op <- par(cex=.75, font=4)
text(-16, c(18,12,3), c("Systematic Allocation", "Random Allocation",
                       "Alternate Allocation"), pos=4)

par(cex=.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"))
text(-16, 20, "Author(s) and Year", pos=4)
text(6, 20, "Relative Risk [95% CI]", pos=2)
par(op)

### see also addpoly.default function for an example where summaries
### for the three subgroups are added to such a forest plot

```

fsn

*Fail-Safe N***Description**

Function to calculate the fail-safe N.

Usage

```
fsn(yi, vi, sei, data, type="Rosenthal",
    alpha=.05, target, subset, digits=4)
```

Arguments

<code>yi</code>	a vector with the observed effect sizes or outcomes.
<code>vi</code>	a vector with the corresponding sampling variances.
<code>sei</code>	a vector with the corresponding standard errors. (note: only one of the two, <code>vi</code> or <code>sei</code> , needs to be specified)
<code>data</code>	an optional data frame containing the variables given to the arguments above.
<code>type</code>	a vector indicating the method to use for the calculation of the fail-safe N. Possible options are "Rosenthal", "Orwin", or "Rosenberg". See below for more details.
<code>alpha</code>	target alpha level to use for the Rosenthal and Rosenberg methods (.05 by default).
<code>target</code>	target average effect size to use for the Orwin method. If undefined, then the target average effect size will be equal to the observed average effect size divided by 2.

subset	an optional vector indicating the subset of studies that should be used for the calculation. This can be a logical vector of length k or a numeric vector indicating the indices of the observations to include.
digits	an integer specifying the number of decimal places to which the printed results should be rounded (default is 4).

Details

The Rosenthal 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 combined significance level (p-value) to a target 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) average effect size to a target (unweighted) average effect size. The method is described in Orwin (1983).

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 significance level (p-value) of the (weighted) average effect size (based on a fixed-effects model) to a target alpha level (e.g., .05). The method is described in Rosenberg (2005).

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 target alpha level.
pval	the p-value of the observed results. NA for the Orwin method.
meanes	the average effect size of the observed results. NA for the Rosenthal method.
target	the target effect size. 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>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Rosenthal, R. (1979). The "file drawer problem" and tolerance for null results. *Psychological Bulletin*, **86**, 638–641.
- Orwin, R. G. (1983). A fail-safe N for effect size in meta-analysis. *Journal of Educational Statistics*, **8**, 157–159.
- Rosenberg, M. S. (2005). The file-drawer problem revisited: A general weighted method for calculating fail-safe numbers in meta-analysis. *Evolution*, **59**, 464–468.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[ranktest](#), [trimfill](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

fsn(yi, vi, data=dat)
fsn(yi, vi, data=dat, type="Orwin")
fsn(yi, vi, data=dat, type="Rosenberg")
```

funnel

Funnel Plots

Description

The function `funnel` is generic. It can be used to create funnel plots.

Usage

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

Arguments

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

Details

Currently, there is only a method for handling objects of class "rma" with the `funnel` function. Accordingly, the corresponding method is called `funnel.rma`. See the documentation for that function for more details.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Light, R. J. & Pillemer, D. B. (1984). *Summing up: The science of reviewing research*. Cambridge, MA: Harvard University Press.
- 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**, 1046–1055.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[funnel.rma](#)

funnel.rma

Funnel Plots for rma Objects

Description

Function to create funnel plots for objects of class "rma".

Usage

```
## S3 method for class 'rma'
funnel(x, xlim=NULL, ylim=NULL, xlab=NULL, ylab="Standard Error",
       steps=5, at=NULL, atranf=FALSE, targ,
       digits=c(2,3), level=x$level, addtau2=FALSE,
       type="rstandard", back="lightgray", shade="white",
       hlines="white", refline=NULL, pch=19, pch.fill=21, ...)
```

Arguments

x	an object of class "rma".
xlim	x axis limits. Defaults to NULL, which means that the function tries to set the x axis limits to some sensible values.
ylim	y axis limits. Defaults to NULL, which means that the function tries to set the y axis limits to some sensible values.
xlab	title for the x axis. If NULL (default), the function tries to set an appropriate axis title.
ylab	title for the y axis.
steps	the number of tick marks for the y axis (default is 5).

at	position of the x axis tick marks and corresponding labels. Defaults to NULL, which means that the function tries to set the tick mark positions/labels to some sensible values.
atransf	an optional argument specifying the name of a function that should be used to transform the x axis labels (e.g., <code>transf=exp</code>). Defaults to FALSE, which means that no transformation is used.
targs	optional arguments needed by the function specified via <code>atransf</code> .
digits	integer value specifying 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 specifying the number of decimal places for the x axis, the second for the y axis labels (default is <code>c(2,3)</code>).
level	numerical value between 0 and 100 specifying the level of the pseudo confidence interval region (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 (default is FALSE). Ignored when the model includes moderators and residuals are plotted.
type	either “ <code>rstandard</code> ” (default) or “ <code>rstudent</code> ” indicating 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.
shade	color to use for shading the pseudo confidence interval region. When <code>level</code> is a vector of values, different shading colors can be specified for each region.
hlines	color of the horizontal reference lines.
refline	value at which the pseudo confidence interval should be centered. Default is NULL, which means that the interval is centered at the fixed- or random-effects model estimate when the model does not include moderators and at zero when moderators are included and residuals are plotted.
pch	plotting symbol to use for the observed effect sizes or outcomes. By default, a solid circle is used. Can also be a vector of values. See points for other options.
pch.fill	plotting symbol to use for the effect sizes or 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.
...	other arguments.

Details

For fixed- and random-effects models (i.e., models not involving moderators), the plot shows the individual 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 $\pm 1.96SE$, where SE is the standard error value from the y axis. If `addtau2=TRUE`, then the bounds of the pseudo confidence interval region are equal to $\pm 1.96\sqrt{SE^2 + \tau^2}$, where τ^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.

If the object passed to the function comes from the `trimfill` function, the effect sizes or outcomes that are filled in by the trim and fill method are also added to the funnel plot.

The arguments `back`, `shade`, and `hlines` can be set to `NULL` to suppress the shading and the horizontal reference lines.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

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**, 991–996.
- 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**, 1046–1055.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

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

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

### standard funnel plot
funnel(res)

### funnel plot with relative risk values on the x axis
funnel(res, attransf=exp, at=log(c(.12, .25, .5, 1, 2)))

### contour-enhanced funnel plot centered at 0 (see Peters et al., 2008)
funnel(res, level=c(90, 95, 99), shade=c("white", "gray", "darkgray"),
       cex=1.2, reflines=0)
```

```
### mixed-effects model with absolute latitude in the model
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=ablat,
          data=dat.bcg, measure="RR", method="REML")
funnel(res)
```

hatvalues.rma.uni *Diagonal Elements of the Hat Matrix for rma.uni Objects*

Description

The function extracts the diagonal elements of the hat matrix for objects of class "rma.uni".

Usage

```
## S3 method for class 'rma.uni'
hatvalues(model, ...)
```

Arguments

model an object of class "rma.uni".
... other arguments.

Value

A vector with the diagonal elements of the hat matrix.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

Viechtbauer, W. & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**, 112–125.

See Also

[rma.uni](#), [influence.rma.uni](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a mixed-effects model
### with two moderators (absolute latitude and publication year)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat, year),
          data=dat.bcg, measure="RR", method="REML")
hatvalues(res)
```

influence.rma.uni *Case Diagnostics for rma.uni Objects*

Description

The function calculates various case diagnostics that indicate the influence of deleting one case at a time on the model fit and the fitted/residual values for objects of class "rma.uni".

Usage

```
## S3 method for class 'rma.uni'
influence(model, digits=model$digits, ...)
```

Arguments

model	an object of class "rma.uni".
digits	an integer specifying 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 following leave-one-out diagnostics are calculated for each study:

- externally standardized residual,
- DFFITS value,
- Cook's distance,
- covariance ratio,
- the leave-one-out amount of (residual) heterogeneity,
- the leave-one-out test statistic for the test of (residual) heterogeneity, and
- DFBETAS value(s).

The diagonal elements of the hat matrix and the weights (in %) given to the observed effects 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 for the i^{th} study changes after excluding the i^{th} study from the model fitting.

Cook's distance can be interpreted as the Mahalanobis distance between the entire set of predicted values once with the i^{th} study included and once with the i^{th} study 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 i^{th} study 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 i^{th} study 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 i^{th} study removed. Note that this is always equal to 0 for fixed-effects models.

Similarly, the leave-one-out test statistic for the test of (residual) heterogeneity is the value of the test statistic of the test for (residual) heterogeneity calculated based on the dataset with the i^{th} study removed.

Finally, the DFBETAS value(s) essentially indicate(s) how many standard deviations the estimated coefficient(s) change(s) after excluding the i^{th} study from the model fitting.

Value

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

inf	A data frame with columns equal to 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, and weights.
dfb	A data frame with columns equal to 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, the 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 present function).

It may not be possible to fit the model after deletion of the i^{th} study from the dataset. This will result in NA values for that study.

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>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

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. <http://www.jstatsoft.org/v36/i03/>.
- Viechtbauer, W. & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**, 112–125.

See Also

[print.infl.rma.uni](#), [plot.infl.rma.uni](#), [rstudent.rma.uni](#), [hatvalues.rma.uni](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a mixed-effects model
### with two moderators (absolute latitude and publication year)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat, year),
           data=dat.bcg, measure="RR", method="REML")
influence(res)
plot(influence(res))
```

 labbe

L'Abbe Plots

Description

The function labbe is generic. It can be used to create L'Abbe plots.

Usage

```
labbe(x, ...)
```

Arguments

x an object of class "rma".
 ... other arguments.

Details

Currently, there is only a method for handling objects of class "rma" with the labbe function. Accordingly, the corresponding method is called `labbe.rma`. See the documentation for that function for more details.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- L'Abbe, K. A., Detsky, A. S., & O'Rourke, K. (1987). Meta-analysis in clinical research. *Annals of Internal Medicine*, **107**, 224–233.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[labbe.rma](#)

labbe.rma

L'Abbe Plots for rma Objects

Description

Function to create L'Abbe plots for objects of class "rma".

Usage

```
## S3 method for class 'rma'
labbe(x, xlim=NULL, ylim=NULL, xlab=NULL, ylab=NULL,
      add=x$add, to=x$to, transf=FALSE, targs, pch=21,
      psize=NULL, bg="gray", ...)
```

Arguments

- | | |
|------|--|
| x | an object of class "rma". See 'Details'. |
| xlim | x axis limits. Defaults to NULL, which means that the function tries to set the x axis limits to some sensible values. |
| ylim | y axis limits. Defaults to NULL, which means that the function tries to set the y axis limits to some sensible values. |
| xlab | title for the x axis. If NULL (default), the function tries to set an appropriate axis title. |
| ylab | title for the y axis. |

add	See below and the documentation of the escalc function for more details.
to	See below and the documentation of the escalc function for more details.
transf	an optional argument specifying the name of a function that should be used to transform the outcomes (e.g., transf=exp). Defaults to FALSE, which means that 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	an optional vector with point sizes for the outcomes. Default is NULL, which means that the point sizes are drawn proportional to the precision of the estimates.
bg	color to use for filling the plotting symbol (default is "gray"). Can also be a vector of values. Set to NA to make the plotting symbols transparent.
...	other arguments.

Details

The model specified by `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](#), or [rma.peto](#) functions. Moreover, the model must be fitted with measure set equal to "RD" (for risk differences), "RR" (for relative risks), "OR" (for odds ratios), "AS" (for arcsine 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 experimental groups (e.g., treatment and control groups) 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) relative risks, the log odds when analyzing (log) odds ratios, the arcsine transformed proportions when analyzing arcsine 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 (for example, `transf=exp` to transform the log of the proportions back to raw proportions).

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 2x2 tables is a common solution to this problem. By default, the functions adopts the same method for handling zero cells as was done when fitting the model.

The size of the points is drawn proportional to the precision (inverse standard error) of the outcomes. 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.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- L'Abbe, K. A., Detsky, A. S., & O'Rourke, K. (1987). Meta-analysis in clinical research. *Annals of Internal Medicine*, **107**, 224–233.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.uni](#), [rma.mh](#), [rma.peto](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

### default plot
labbe(res)

### funnel plot with risk values on the x and y axis
labbe(res, transf=exp)
```

leave1out

Leave-One-Out Diagnostics

Description

The function `leave1out` is generic. For suitable model objects, it repeatedly fits the model, leaving out one observation at a time.

Usage

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

Arguments

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

Details

Currently, there are methods for handling objects of class `"rma.uni"`, `"rma.mh"`, and `"rma.peto"` with the `leave1out` function. Accordingly, the corresponding methods are called `leave1out.rma.uni`, `leave1out.rma.mh`, and `leave1out.rma.peto`. See the documentation for those functions for more details.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

Viechtbauer, W. & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**, 112–125.

See Also

[leave1out.rma.uni](#), [leave1out.rma.mh](#), [leave1out.rma.peto](#)

leave1out.rma.mh

Leave-One-Out Diagnostics for rma.mh and rma.peto Objects

Description

The functions `leave1out.rma.mh` and `leave1out.rma.peto` repeatedly fit the specified model, leaving out one observation (i.e., 2x2 table) at a time.

Usage

```
## S3 method for class 'rma.mh'
leave1out(x, digits=x$digits, transf=FALSE, ...)
## S3 method for class 'rma.peto'
leave1out(x, digits=x$digits, transf=FALSE, ...)
```

Arguments

<code>x</code>	an object of class "rma.mh" or "rma.peto".
<code>digits</code>	an integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
<code>transf</code>	logical indicating whether odds ratios or relative risks (and the corresponding confidence interval bounds) should be given in the transformed (meaning: raw) units or in terms of log units (the default).
<code>...</code>	other arguments.

Value

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

estimate	estimated coefficients of the model.
se	standard errors of the coefficients. NA if transf=TRUE.
zval	test statistics of the coefficients.
pval	p-values for the test statistics.
ci.lb	lower bounds of the confidence intervals for the coefficients.
ci.ub	upper bounds of the confidence intervals for the coefficients.
Q	test statistics for the tests of heterogeneity.
Qp	p-values for the tests of heterogeneity.

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

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.
- Viechtbauer, W. & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**, 112–125.

See Also

[leave1out](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

leave1out(res)
leave1out(res, 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)

leave1out(res)
leave1out(res, transf=TRUE)
```

leave1out.rma.uni *Leave-One-Out Diagnostics for rma.uni Objects*

Description

The function `leave1out.rma.uni` repeatedly fits the specified model, leaving out one observation at a time.

Usage

```
## S3 method for class 'rma.uni'
leave1out(x, digits=x$digits, transf=FALSE, targs, ...)
```

Arguments

<code>x</code>	an object of class "rma.uni".
<code>digits</code>	an integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
<code>transf</code>	an optional argument specifying the name of a function that should be used to transform the model coefficients and interval bounds (e.g., <code>transf=exp</code>). Defaults to <code>FALSE</code> , which means that no transformation is used.
<code>targs</code>	optional arguments needed by the function specified under <code>transf</code> .
<code>...</code>	other arguments.

Details

The model specified by `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:

<code>estimate</code>	estimated coefficients of the model.
<code>se</code>	standard errors of the coefficients. NA if <code>transf</code> is used to transform the coefficients.
<code>zval</code>	test statistics of the coefficients.
<code>pval</code>	p-values for the test statistics.
<code>ci.lb</code>	lower bounds of the confidence intervals for the coefficients.
<code>ci.ub</code>	upper bounds of the confidence intervals for the coefficients.
<code>Q</code>	test statistics for the tests of heterogeneity.
<code>Qp</code>	p-values for the tests of heterogeneity.
<code>tau2</code>	estimated amounts of (residual) heterogeneity (only for random-effects models).
<code>I2</code>	values of I^2 (only for random-effects models).
<code>H2</code>	values of H^2 (only for random-effects models).

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

Note

Various case diagnostics for objects of class "rma.uni" can also be obtained with the `influence.rma.uni` function.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.
- Viechtbauer, W. & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**, 112–125.

See Also

[leave1out](#), [influence.rma.uni](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

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

leave1out(res)
leave1out(res, transf=exp)
```

metafor.news

Read News File of the Metafor Package

Description

Read news file of the [metafor-package](#) package.

Usage

```
metafor.news(...)
```

Arguments

... arguments passed to `file.show`.

Details

The function just makes use of the `file.show` command to display the ‘NEWS’ file of the package.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[file.show](#)

Examples

```
metafor.news()
```

permutest

Permutation Tests

Description

The function `permutest` is generic. It can be used to carry out permutation tests for specific classes of objects.

Usage

```
permutest(x, ...)
```

Arguments

x an object for which permutation tests can be carried out. See ‘Details’.
... other arguments.

Details

Currently, there is only a method for objects of class “`rma.uni`” created by the `rma.uni` function. Accordingly, the corresponding method is called `permutest.rma.uni`. See the documentation for that function for more details.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

Good, P. I. (2009). *Permutation, parametric, and bootstrap tests of hypotheses* (3rd ed.). New York: Springer.

See Also

[permutest.rma.uni](#)

permutest.rma.uni *Permutation Tests for rma.uni Objects*

Description

The function carries out permutation tests for objects of class "rma.uni".

Usage

```
## S3 method for class 'rma.uni'
permutest(x, exact=FALSE, iter=1000, progbar=TRUE,
          retpermdist=FALSE, digits=x$digits, ...)
```

Arguments

x	an object of class "rma.uni".
exact	logical indicating whether an exact permutation test should be carried out or not (default is FALSE). See 'Details'.
iter	integer to specify the number of iterations for the permutation test when not doing an exact test (default is 1000 iterations).
progbar	logical indicating whether a progress bar should be shown (default is TRUE).
retpermdist	logical indicating whether the permutation distributions of the test statistics should also be returned (default is FALSE).
digits	an integer specifying 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

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 two times the proportion of times that 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 design matrix. The (two-sided) p-value for a particular model coefficient is then equal to two times the proportion of times that 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) 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), especially with respect to the possible number of permutations of the design matrix. For example, for $k = 5$, there are only 120 possible permutations of the design matrix (32 possible permutations of the signs). For $k = 8$, there are already 40,320 (256). And for $k = 10$, there are 3,628,800 (1024). Therefore, going through all possible permutations may become infeasible.

Instead of using an exact permutation test, one can set `exact=FALSE` (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 will yield (slightly) different p-values. Setting `iter` sufficiently large ensures that the results become stable.

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.

Also, the function will terminate with an error if the number of iterations is too large.

Finally, it is important to note that 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, k must be at least equal to 6 to reject the null hypothesis at $\alpha = .05$. For models with moderators, the smallest possible (two-sided) p-value for a particular model coefficient is .08333 when $k = 4$ and .01667 when $k = 5$. Therefore, k must be at least equal to 5 to reject the null hypothesis at $\alpha = .05$.

Value

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

<code>pval</code>	p-value(s) based on the permutation test.
<code>QMp</code>	p-value for the omnibus test of coefficients based on the permutation test.
<code>zval.perm</code>	values of the test statistics of the coefficients under the various permutations (only when <code>retpermdist=TRUE</code>).
<code>QM.perm</code>	values of the test statistic for the omnibus test of coefficients under the various permutations (only when <code>retpermdist=TRUE</code>).

... 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.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Follmann, D. A. & Proschan, M. A. (1999). Valid inference in random effects meta-analysis. *Biometrics*, **55**, 732–737.
- 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**, 1663–1682.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.uni](#), [print.permutest.rma.uni](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

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

### 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=cbind(ablat, year), data=dat, method="REML")

### 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$zval.perm[,2], breaks=120, freq=FALSE, xlim=c(-5,5), ylim=c(0,.4),
     main="Permutation Distribution", xlab="Value of Test Statistic")
abline(v=res$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.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, plotdfb=FALSE, dfbnew=FALSE, logcov=TRUE, las=0,
     pch=21, bg="black", bg.infl="red", col.na="lightgray", ...)
```

Arguments

x	an object of class "infl.rma.uni".
plotdfb	logical indicating whether the DFBETAS values should be plotted (default is FALSE).
dfbnew	logical indicating whether a new device should be opened for plotting the DFBETAS values (default is FALSE). If set to TRUE, then plotdfb is automatically set to TRUE as well.
logcov	logical indicating whether the covariance ratios should be plotted on a log scale (default is TRUE).
las	numeric (0,1,2,3). Defines style of 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 (default is "black").
bg.infl	color to use for filling the plotting symbol when the point is considered influential (default is "red").
col.na	color to use for lines connecting two points with NAs in between (default is "lightgray").
...	other arguments.

Details

The function plots 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, and weights. If `plotdfb=TRUE`, the DFBETAS values are also plotted either after confirming the page change (if `newdfb=FALSE`) or on a separate device (if `newdfb=TRUE`). Points are considered 'influential' if at least one of the following is true:

- The absolute DFFITS value is larger than $3\sqrt{p/(k-p)}$, where p is the number of model coefficients and k the number of studies.
- 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 $3(p/k)$.
- Any DFBETAS value is larger than 1.

These cut-offs are also 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 studies. 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 studies. 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 study on the results.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.
- Viechtbauer, W. & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**, 112–125.

See Also

[influence.rma.uni](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a mixed-effects model
### with two moderators (absolute latitude and publication year)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat, year),
           data=dat.bcg, measure="RR", method="REML")
plot(influence(res))
plot(influence(res), plotdfb=TRUE)
```

plot.rma.uni

Plot Method for rma Objects

Description

Plot method for objects of class "rma.uni", "rma.mh", and "rma.peto".

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, ...)
```

Arguments

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

qqplot logical indicating whether a normal QQ plot should be drawn (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 quite impossible to read.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[forest](#), [funnel](#), [radial](#), [qqnorm.rma.uni](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a random-effects model
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg,
           measure="RR", method="REML")
plot(res, qqplot=TRUE)

### mixed-effects model with two moderators (absolute latitude and publication year)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat, year),
           data=dat.bcg, measure="RR", method="REML")
plot(res, qqplot=TRUE)
```

predict.rma

Predicted Values for rma Objects

Description

The function calculates predicted values, corresponding standard errors, confidence intervals, and (approximate) credibility intervals for objects of class "rma".

Usage

```
## S3 method for class 'rma'
predict(object, newmods, addx=FALSE, level=object$level,
        digits=object$digits, transf=FALSE, targs, ...)
```

Arguments

object	an object of class "rma".
newmods	an optional vector or matrix specifying the values of the moderator values for which the predicted values should be calculated. See 'Details'.
addx	logical, specifying whether the values of the moderator variables should be added to the returned object. See 'Examples'.
level	a numerical value between 0 and 100 specifying the confidence and credibility interval level (the default is to take the value from the object).
digits	an integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
transf	an optional argument specifying the name of a function that should be used to transform the predicted values and interval bounds (e.g., transf=exp). Defaults to FALSE, which means that no transformation is used.
targs	optional arguments needed by the function specified under transf.
...	other arguments.

Details

For the 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.

For the 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.

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 q moderator variables, new moderator values for k_{new} new studies can be specified by setting `newmods` equal to an $k_{new} \times q$ matrix with the corresponding new moderator values. Examples are shown below.

For random/mixed-effects models, an approximate credibility interval is also calculated. The interval estimates where `level` % of the true outcomes fall in the hypothetical population of studies. Note that this interval is calculated under the assumption that the value of τ^2 is known (and not estimated). A proper method for calculating a credibility interval that accounts for the uncertainty in the estimate of τ^2 will be implemented in the future.

Value

An object of class "predict.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.lb	upper bound of the confidence interval(s).
cr.lb	lower bound of the credibility interval(s) (only random/mixed-effects models).
cr.ub	upper bound of the credibility interval(s) (only random/mixed-effects models).
X	the moderator value(s) used to calculate the predicted values (only when addx=TRUE).
...	some additional elements/values.

The "predict.rma" object is formatted and printed with `print.predict.rma`.

Note

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` (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 set equal to NA.

The normal distribution is used to calculate the confidence/credibility intervals. When the model was fitted with the Knapp and Hartung (2003) method (i.e., `knha=TRUE` in the `rma.uni` function), then the t-distribution with $k - p$ degrees of freedom is used.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Hedges, L. V. & Olkin, I. (1985). *Statistical methods for meta-analysis*. San Diego, CA: Academic Press.
- Raudenbush, S. W. (2009). Analyzing effect sizes: Random effects models. In H. C. 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. <http://www.jstatsoft.org/v36/i03/>.

See Also

`fitted.rma`, `blup.rma.uni`

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

```

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

### mixed-effects model with absolute latitude as a moderator
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat),
          measure="RR", data=dat.bcg, method="REML")

### predicted average relative risks for given absolute latitude values
predict(res, transf=exp, addx=TRUE)

### predicted average relative risks for 10-60 degrees absolute latitude
predict(res, newmods=c(10, 20, 30, 40, 50, 60), transf=exp)

### mixed-effects model with two moderators (absolute latitude and publication year)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat, year),
          measure="RR", data=dat.bcg, method="REML")

### predicted average relative risks 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)

```

print.anova.rma.uni *Print Method for anova.rma.uni Objects*

Description

Print method for objects of class "anova.rma.uni".

Usage

```
## S3 method for class 'anova.rma.uni'
print(x, digits=x$digits, ...)
```

Arguments

x	an object of class "anova.rma.uni".
digits	an integer specifying 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, and log likelihood of the full and the reduced model.
- the value of the likelihood ratio test statistic.
- the p-value for the likelihood ratio test.

- the test statistic for the test of (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.
- the amount of (residual) heterogeneity in the reduced model that is accounted for in the full model (in percent). NA for fixed-effects models or if the amount of heterogeneity in the reduced model is equal to zero.

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[anova.rma.uni](#)

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	an integer specifying 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>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[fsn](#)

print.infl.rma.uni *Print Method for infl.rma.uni Objects*

Description

Print method for objects of class "infl.rma.uni".

Usage

```
## S3 method for class 'infl.rma.uni'  
print(x, digits=x$digits, ...)
```

Arguments

x	an object of class "infl.rma.uni".
digits	an integer specifying 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 [influence.rma.uni](#) function for details on what is printed.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

Viechtbauer, W. & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**, 112–125.

See Also

[influence.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	an integer specifying 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.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

```
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.legend=TRUE, ...)
```

Arguments

x	an object of class "permutest.rma.uni".
digits	an integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
signif.legend	logical indicating whether the legend for the 'significance stars' should be printed.
...	other arguments.

Details

The output includes:

- the results of the omnibus test of the coefficients in the model. 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 test are indicated by pval*.

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[permutest.rma.uni](#)

print.predict.rma *Print Method for predict.rma Objects*

Description

Print method for objects of class "predict.rma".

Usage

```
## S3 method for class 'predict.rma'  
print(x, digits=x$digits, ...)
```

Arguments

x	an object of class "predict.rma".
digits	an integer specifying 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 [predict.rma](#) function for details on what is printed.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

print.ranktest.rma *Print Method for ranktest.rma Objects*

Description

Print method for objects of class "ranktest.rma".

Usage

```
## S3 method for class 'ranktest.rma'  
print(x, digits=x$digits, ...)
```

Arguments

x	an object of class "ranktest.rma".
digits	an integer specifying 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>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[ranktest.rma](#)

print.regtest.rma *Print Method for regtest.rma Objects*

Description

Print method for objects of class "regtest.rma".

Usage

```
## S3 method for class 'regtest.rma'  
print(x, digits=x$digits, ...)
```

Arguments

x	an object of class "regtest.rma".
digits	an integer specifying 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 model used for the regression test
- the predictor used for the regression test
- the value of the test statistic for the test that the predictor is unrelated to the outcomes
- the degrees of freedom of the test statistic (only if the test statistic follows a t-distribution)
- the p-value for the test statistic

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[regtest.rma](#)

print.rma.uni *Print Methods for rma Objects*

Description

Print methods for objects of class "rma.uni", "rma.mh", and "rma.peto".

Usage

```
## S3 method for class 'rma.uni'
print(x, digits=x$digits, showfit=FALSE, signif.legend=TRUE, ...)
## S3 method for class 'rma.mh'
print(x, digits=x$digits, showfit=FALSE, ...)
## S3 method for class 'rma.peto'
print(x, digits=x$digits, showfit=FALSE, ...)
## S3 method for class 'rma'
summary(object, digits=object$digits, showfit=TRUE, signif.legend=TRUE, ...)
```

Arguments

x	an object of class "rma.uni", "rma.mh", or "rma.peto" (for print).
object	an object of class "rma" (for summary).
digits	an integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
showfit	logical indicating whether the fit statistics and information criteria should be printed.
signif.legend	logical indicating whether the legend for the 'significance stars' should be printed.
...	other arguments.

Details

The output includes:

- the log likelihood, deviance, AIC, and BIC value (when `showfit=TRUE`).
- 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. For ML and REML estimation, the asymptotic standard error of the estimate of τ^2 is also provided.
- the I^2 statistic estimates (in percent) how much of the total variability in the effect size estimates (which is composed of heterogeneity and sampling variability) can be attributed to heterogeneity among the true effects. Only supplied when fitting a random-effects model.
- the H^2 statistic is the ratio of the total amount of variability in the observed outcomes to the amount of sampling variability. Only supplied when fitting a random-effects model.
- 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 QE-test for residual heterogeneity when moderators are included.

- the results of the omnibus 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.

When analyzing odds ratios using the Mantel-Haenszel method, the Cochran-Mantel-Haenszel test and Tarone's test for heterogeneity are also provided.

Value

The function does not return an object.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.uni](#), [rma.mh](#), [rma.peto](#)

qqnorm.rma.uni	Normal QQ Plot 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, ...)
## S3 method for class 'rma.mh'
qqnorm(y, type="rstandard", pch=19, ...)
## S3 method for class 'rma.peto'
qqnorm(y, type="rstandard", pch=19, ...)
```

Arguments

<code>y</code>	an object of class "rma.uni", "rma.mh", or "rma.peto".
<code>type</code>	either "rstandard" (default) or "rstudent" indicating whether the usual or deleted residuals should be used in creating the plot. See 'Details'.
<code>pch</code>	plotting symbol to use for the observed effect sizes or outcomes. By default, a solid circle is used. See points for other options.
<code>envelope</code>	logical indicating whether a pseudo confidence envelope should be simulated and added to the plot (default is TRUE). Only for objects of class "rma.uni". See 'Details'.
<code>level</code>	numerical value between 0 and 100 specifying the level of the pseudo confidence envelope (the default is to take the value from the object).
<code>bonferroni</code>	logical indicating whether the bounds of the envelope should be Bonferroni corrected.
<code>reps</code>	numerical value indicating the number of iterations to use for simulating the pseudo confidence envelope (default is 1000).
<code>smooth</code>	logical indicating whether the results from the simulation should be smoothed (default is TRUE).
<code>bass</code>	numerical value that controls the degree of smoothing (default is 0).
<code>...</code>	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.

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.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

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. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.uni](#), [rma.mh](#), [rma.peto](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

### mixed-effects model with absolute latitude as a moderator
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat),
           measure="RR", data=dat.bcg, method="REML")
qqnorm(res)
```

radial

Radial Plots

Description

The function `radial` is generic. It can be used to create radial (also called Galbraith) plots.

Usage

```
radial(x, ...)
galbraith(x, ...)
```

Arguments

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

Details

Currently, there is only a method for handling objects of class "rma" with the `radial` function. Accordingly, the corresponding method is called `radial.rma`. See the documentation for that function for more details.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Galbraith, R. F. (1988). Graphical display of estimates having differing standard errors. *Technometrics*, **30**, 271–281.
- Galbraith, R. F. (1988). A note on graphical presentation of estimated odds ratios from several clinical trials. *Statistics in Medicine*, **7**, 889–894.
- Galbraith, R. F. (1994). Some applications of radial plots. *Journal of the American Statistical Association*, **89**, 1232–1242.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[radial.rma](#)

radial.rma

Radial Plots for rma Objects

Description

Function to create radial (Galbraith) plots for objects of class "rma".

Usage

```
## S3 method for class 'rma'
radial(x, center=FALSE, xlim=NULL, zlim=NULL, xlab=NULL, zlab=NULL,
       atz=NULL, aty=NULL, steps=7, level=x$level, digits=2,
       back="lightgray", transf=FALSE, targs, pch=19,
       arc.res = 100, cex=NULL, ...)
```

Arguments

- | | |
|--------|---|
| x | an object of class "rma". |
| center | logical to indicate whether the plot should be centered horizontally at the model estimate (default is FALSE). |
| xlim | x axis limits. Defaults to NULL, which means that the function tries to set the x axis limits to some sensible values. |
| zlim | z axis limits. Defaults to NULL, which means that 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. Defaults to NULL, which means that a label is generated by the function.
zlab	title for the z axis. Defaults to NULL, which means that a label is generated by the function.
atz	position for the z axis tick marks and labels. Defaults to NULL, which means that these are set by the function.
aty	position for the y axis tick marks and labels. Defaults to NULL, which means that these are set by the function.
steps	the number of tick marks for the y axis (default is 7). Ignored when argument aty is used.
level	a numerical value between 0 and 100 specifying the level of the z axis error region (the default is to take the value from the object).
digits	integer value specifying the number of decimal places to which the tick mark labels of the y axis should be rounded (default is 2).
back	color of the z axis error region. Set to NA to suppress shading of the region.
transf	an optional argument specifying the name of a function that should be used to transform the y axis labels (e.g., transf=exp). Defaults to FALSE, which means that no transformation is used.
targs	optional arguments needed by the function specified via transf.
pch	plotting symbol. By default, a solid circle is used. See points for other options.
arc.res	integer value specifying the number of line segments to use when drawing the y axis and confidence interval arcs (default is 100).
cex	an optional character and symbol expansion factor. If NULL (default), 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 individual 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 individual 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 individual 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 individual observed effect sizes or outcomes.

If the model contains moderators, the function returns an error.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Galbraith, R. F. (1988). Graphical display of estimates having differing standard errors. *Technometrics*, **30**, 271–281.
- Galbraith, R. F. (1988). A note on graphical presentation of estimated odds ratios from several clinical trials. *Statistics in Medicine*, **7**, 889–894.
- Galbraith, R. F. (1994). Some applications of radial plots. *Journal of the American Statistical Association*, **89**, 1232–1242.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.uni](#), [rma.mh](#), [rma.peto](#),

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a fixed-effects model
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg,
          data=dat.bcg, measure="RR", method="FE")
radial(res)

### line from (0,0) with slope equal to the log relative risk from the 4th study
abline(a=0, b=c(-1.44155119), lty="dotted")

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

ranktest

Rank Correlation Test for Funnel Plot Asymmetry

Description

The function `ranktest` is generic. It can be used to carry out the rank correlation test for funnel plot asymmetry as described by Begg and Mazumdar (1994).

Usage

```
ranktest(x, ...)
```

Arguments

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

Details

Currently, there is only a method for handling objects of class "rma" with the ranktest function. Accordingly, the corresponding method is called `ranktest.rma`. See the documentation for that function for more details.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

- Begg, C. B. & Mazumdar, M. (1994). Operating characteristics of a rank correlation test for publication bias. *Biometrics*, **50**, 1088–1101.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[ranktest.rma](#)

ranktest.rma

Rank Correlation Test for Funnel Plot Asymmetry for rma Objects

Description

Rank correlation test for funnel plot asymmetry for objects of class "rma".

Usage

```
## S3 method for class 'rma'  
ranktest(x, ...)
```

Arguments

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

Value

An object of class "ranktest.rma". 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.rma` function.

Note

The method does not depend on the model fitted. Therefore, regardless of the model, 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>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Begg, C. B. & Mazumdar, M. (1994). Operating characteristics of a rank correlation test for publication bias. *Biometrics*, **50**, 1088–1101.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[ranktest](#), [regtest](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

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

ranktest(res)
```

regtest

Regression Tests for Funnel Plot Asymmetry

Description

The function `regtest` is generic. It can be used to carry out various tests for funnel plot asymmetry, including Egger's regression test and variations thereof.

Usage

```
regtest(x, ...)
```

Arguments

<code>x</code>	an object of class "rma".
<code>...</code>	other arguments.

Details

Currently, there is only a method for handling objects of class "rma" with the `regtest` function. Accordingly, the corresponding method is called `regtest.rma`. See the documentation for that function for more details.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

- Begg, C. B. & Mazumdar, M. (1994). Operating characteristics of a rank correlation test for publication bias. *Biometrics*, **50**, 1088–1101.
- Egger, M., Davey Smith, G., Schneider, M., & Minder, C. (1997). Bias in meta-analysis detected by a simple, graphical test. *British Medical Journal*, **315**, 629–634.
- 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**, 3443–3457.
- Macaskill, P., Walter, S. D., & Irwig, L. (2001). A comparison of methods to detect publication bias in meta-analysis. *Statistics in Medicine*, **20**, 641–654.
- 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.
- 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**, 676–680.

Ruecker, G., Schwarzer, G., & Carpenter, J. (2008). Arcsine test for publication bias in meta-analyses with binary outcomes. *Statistics in Medicine*, **27**, 746–763.

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.) (2005) *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. <http://www.jstatsoft.org/v36/i03/>.

See Also

[regtest.rma](#)

regtest.rma

Regression Tests for Funnel Plot Asymmetry for rma Objects

Description

The function can be used to carry out various tests for funnel plot asymmetry, including Egger's regression test and variations thereof, for objects of class "rma".

Usage

```
## S3 method for class 'rma'
regtest(x, model="rma", predictor="sei", ni, ...)
```

Arguments

x	an object of class "rma".
model	either "rma" or "lm" to indicate the type of model to use for the regression test. See 'Details'.
predictor	either "sei" "vi", "ni", or "ninv" to indicate the type of independent variable to use for the regression test. See 'Details'.
ni	an optional vector with the total sample sizes of the studies. This argument only needs to be specified when predictor="ni" or predictor="ninv" and the object x does not already contain the sample size information. See 'Details'.
...	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), Ruecker, Schwarzer, and Carpenter (2008), and Moreno et al. (2009). The various versions of the regression test differ in terms of the model (either a regular weighted regression with a multiplicative dispersion term or one of the meta-analytic models is

used), in terms of the independent variable that the observed outcomes are hypothesized to be related to when publication bias is present (suggested predictors include the standard error, the sampling variance, the total sample size, and the inverse of the total sample size), and in terms of the outcome measure used (e.g., for 2x2 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 outcomes and the chosen predictor, then this usually implies asymmetry in the funnel plot and may be an indication of publication bias.

The `regtest.rma` 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 `knha` 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-effects model, one should first fit a model with, for example, `method="REML"` and then use the `regtest.rma` function.

The predictor is chosen via the `predictor` argument, with `predictor="sei"` for the standard error, `predictor="vi"` for the sampling variance, `predictor="ni"` for the total sample size, and `predictor="ninv"` for the inverse of the total sample size. The object `x` will contain information about the total sample sizes when measure was *not* equal to "GEN" during the initial model fitting.

Finally, depending on what outcome measure was used for the model fitting will determine which outcome measure is used for the regression test.

Value

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

<code>type</code>	the type of test.
<code>model</code>	the model used for the regression test.
<code>predictor</code>	the predictor used for the regression test.
<code>zval</code>	the value of the test statistic.
<code>pval</code>	the corresponding p-value
<code>dfs</code>	the degrees of freedom of the test statistic (if the test is based on a t-distribution).

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

Note

All of the tests do not directly test for publication bias, but for a relationship between the observed outcomes and the chosen predictor. If such a relationship is present, then this usually implies asymmetry in the funnel plot and 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>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Begg, C. B. & Mazumdar, M. (1994). Operating characteristics of a rank correlation test for publication bias. *Biometrics*, **50**, 1088–1101.
- Egger, M., Davey Smith, G., Schneider, M., & Minder, C. (1997). Bias in meta-analysis detected by a simple, graphical test. *British Medical Journal*, **315**, 629–634.
- 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**, 3443–3457.
- Macaskill, P., Walter, S. D., & Irwig, L. (2001). A comparison of methods to detect publication bias in meta-analysis. *Statistics in Medicine*, **20**, 641–654.
- 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.
- 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**, 676–680.
- Ruecker, G., Schwarzer, G., & Carpenter, J. (2008). Arcsine test for publication bias in meta-analyses with binary outcomes. *Statistics in Medicine*, **27**, 746–763.
- 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.) (2005) *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. <http://www.jstatsoft.org/v36/i03/>.

See Also

[regtest](#), [ranktest](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

### fixed-effects model
res <- rma(yi, vi, data=dat, method="FE")
regtest(res)

### more examples
res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg,
          data=dat, method="REML")
regtest(res, model="lm", predictor="ni")

res <- rma(measure="PETO", ai=tpos, bi=tneg, ci=cpos, di=cneg,
          data=dat, method="FE")
```

```

regtest(res, predictor="ninv")

res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg,
          data=dat, mods=cbind(ablat), method="REML")
regtest(res, predictor="ninv")

```

residuals.rma

*Residual Values based on rma Objects***Description**

The `residuals`, `rstandard`, and `rstudent` functions can be used to extract residuals, corresponding standard errors, and standardized residuals for models fitted with the `rma.uni`, `rma.mh`, and `rma.peto` functions.

Usage

```

## S3 method for class 'rma'
residuals(object, ...)
## S3 method for class 'rma.uni'
rstandard(model, digits=model$digits, ...)
## S3 method for class 'rma.mh'
rstandard(model, digits=model$digits, ...)
## S3 method for class 'rma.peto'
rstandard(model, digits=model$digits, ...)
## S3 method for class 'rma.uni'
rstudent(model, digits=model$digits, ...)
## S3 method for class 'rma.mh'
rstudent(model, digits=model$digits, ...)
## S3 method for class 'rma.peto'
rstudent(model, digits=model$digits, ...)

```

Arguments

<code>object</code>	an object of class "rma" (for residuals).
<code>model</code>	an object of class "rma.uni", "rma.mh", or "rma.peto" (for <code>rstandard</code> and <code>rstudent</code>).
<code>digits</code>	an integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
<code>...</code>	other arguments.

Details

The observed residuals (obtained with `residuals`) are simply equal to the 'observed - fitted' values. Dividing the observed residuals by their corresponding standard errors yields (internally) standardized residuals. These can be obtained with `rstandard`.

The `rstudent` function calculates externally standardized residuals (studentized deleted residuals). The externally standardized residual for the i^{th} case is obtained by deleting the i^{th} case from the dataset, fitting the model based on the remaining cases, calculating the predicted value for the i^{th} case based on the fitted model, taking the difference between the observed and the predicted value for the i^{th} case (the deleted residual), and then standardizing the deleted residual. The standard error of the deleted residual is equal to the square root of the sampling variance of the i^{th} case plus the variance of the predicted value plus the amount of (residual) heterogeneity from the fitted model (for fixed-effects models, this last part is always equal to zero).

If a particular study fits the model, its standardized residual follows (asymptotically) a standard normal distribution. A large standardized residual for a study therefore may suggest that the study does not fit the assumed model (i.e., it may be an outlier).

See also `influence.rma.uni` for other leave-one-out diagnostics that are useful for detecting influential cases in models fitted with the `rma.uni` function.

Value

Either a vector with the observed residuals (for `residuals`) or an object of class `"list.rma"`, which is a list containing the following components:

<code>resid</code>	observed residuals (for <code>rstandard</code>) or deleted residuals (for <code>rstudent</code>).
<code>se</code>	corresponding standard errors.
<code>z</code>	standardized residuals (internally standardized for <code>rstandard</code> or externally standardized for <code>rstudent</code>).

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

Note

Right now, the externally standardized residuals are calculated by refitting the model k times. Depending on how large k is, it may take a few moments to finish the calculations.

It may not be possible to fit the model after deletion of the i^{th} case from the dataset. This will result in NA values for that case when calling `rstudent`.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

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. <http://www.jstatsoft.org/v36/i03/>.
- Viechtbauer, W. & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, **1**, 112–125.

See Also

[rma.uni](#), [rma.mh](#), [rma.peto](#), [influence.rma.uni](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

### mixed-effects model with absolute latitude as a moderator
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat),
          measure="RR", data=dat.bcg, method="REML")
rstudent(res)
```

rma.mh

Meta-Analysis via the Mantel-Haenszel Method

Description

Function to fit a fixed-effects model via the Mantel-Haenszel method.

Usage

```
rma.mh(ai, bi, ci, di, n1i, n2i,
       x1i, x2i, t1i, t2i, measure="OR",
       data, slab, subset,
       add=c(1/2,0), to=c("only0","none"),
       level=95, digits=4)
```

Arguments

ai	vector to specify the 2x2 table frequencies (upper left cell). See below and the documentation of the escalc function for more details.
bi	vector to specify the 2x2 table frequencies (upper right cell). See below and the documentation of the escalc function for more details.
ci	vector to specify the 2x2 table frequencies (lower left cell). See below and the documentation of the escalc function for more details.
di	vector to specify the 2x2 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.

<code>x1i</code>	vector to specify the number of cases (first group). See below and the documentation of the <code>escalc</code> function for more details.
<code>x2i</code>	vector to specify the number of cases (second group). See below and the documentation of the <code>escalc</code> function for more details.
<code>t1i</code>	vector to specify the total person-times (first group). See below and the documentation of the <code>escalc</code> function for more details.
<code>t2i</code>	vector to specify the total person-times (second group). See below and the documentation of the <code>escalc</code> function for more details.
<code>measure</code>	a character string indicating the outcome measure to use for the meta-analysis: the odds ratio ("OR"), relative risk ("RR"), risk difference ("RD"), or the incidence rate ratio ("IRR").
<code>data</code>	an optional data frame containing the data supplied to the function.
<code>slab</code>	an optional vector with unique labels for the k studies.
<code>subset</code>	an optional vector indicating the subset of tables that should be used for the analysis. This can be a logical vector of length k or a numeric vector indicating the indices of the tables to include.
<code>add</code>	vector with two non-negative numbers indicating the amount to add to zero cells, counts, or frequencies. The first number is used in the calculation of the individual outcomes, the second number is used when applying the Mantel-Haenszel method. See below and the documentation of the <code>escalc</code> function for more details.
<code>to</code>	a character vector with two strings indicating when the values under <code>add</code> should be added (either "all", "only0", "if0all", or "none"). The first string again applies when calculating the individual outcomes, the second string when applying the Mantel-Haenszel method. See below and the documentation of the <code>escalc</code> function for more details.
<code>level</code>	a numerical value between 0 and 100 specifying the confidence interval level (default is 95).
<code>digits</code>	an integer specifying the number of decimal places to which the printed results should be rounded (default is 4).

Details

When the outcome measure is either the odds ratio (`measure="OR"`), relative risk (`measure="RR"`), or risk difference (`measure="RD"`), the studies are assumed to provide data in terms of 2x2 tables of the form:

	outcome 1	outcome 2	total
group 1	a_i	b_i	n_{1i}
group 2	c_i	d_i	n_{2i}

where a_i , b_i , c_i , and d_i denote the cell frequencies and n_{1i} and n_{2i} 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 a_i , b_i , c_i , and d_i or alternatively a_i , c_i , $n1_i$, and $n2_i$.

Alternatively, when the outcome measure is the incidence rate ratio (measure="IRR"), the studies are assumed to provide data in terms of tables of the form:

	cases	person-time
group 1	$x1_i$	$t1_i$
group 2	$x2_i$	$t2_i$

where $x1_i$ and $x2_i$ denote the number of cases in the first and the second group, respectively, and $t1_i$ and $t2_i$ the corresponding total person-times at risk.

An approach for aggregating table data of these types was suggested by Mantel and Haenszel (1959). The Mantel-Haenszel method provides a weighted estimate under a fixed-effects model. The method is particularly advantageous when aggregating a large number of tables with small sample sizes (the so-called sparse data or increasing strata case). When analyzing odds ratios, the Cochran-Mantel-Haenszel test and Tarone's test for heterogeneity are also provided. When analyzing odds ratios, relative risks, or incidence rate ratios, the printed results are given both in terms of the log and the raw units (for easier interpretation).

The Mantel-Haenszel method itself does not require the calculation of the individual outcome values and directly makes use of the table counts. Zero cells (or zero cases in one or both groups) are not a problem (except in extreme cases, such as when one of the two outcomes never occurs in any of the 2x2 tables or when there are no cases for one of the two groups in any of the tables). Therefore, it is usually unnecessary to add some constant to the cell counts (or the number of cases) when there are zero cells. However, for plotting and various other functions, it is necessary to calculate the individual outcome values for the k tables. Here, zero cells (or zero cases in one or both groups) can be problematic, so adding a constant value to the cell counts (or the number of cases) ensures that all k values can be calculated. The `add` and `to` arguments are used to specify what value should be added to the 2x2 cell frequencies (or the number of cases) and under what circumstances when calculating the individual outcome values and when applying the Mantel-Haenszel method. The documentation of the `escalc` function explains how the `add` and `to` arguments should be used. The first value of the `add` and `to` arguments applies when calculating the individual outcomes, the second value when applying the Mantel-Haenszel method.

Value

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

<code>b</code>	aggregated log odds ratio, log relative risk, or risk difference.
<code>se</code>	standard error of the aggregated value.
<code>zval</code>	test statistics of the aggregated value.
<code>pval</code>	p-value for the test statistic.
<code>ci.lb</code>	lower bound of the confidence interval.
<code>ci.ub</code>	upper bound of the confidence interval.
<code>QE</code>	test statistic for the test of heterogeneity.
<code>QEp</code>	p-value for the test of heterogeneity.

CMH	Cochran-Mantel-Haenszel test statistic (only when measure="OR").
CMHp	corresponding p-value (only when measure="OR").
TA	Tarone's heterogeneity test statistic (only when measure="OR").
TAp	corresponding p-value (only when measure="OR").
k	number of tables included in the analysis.
yi, vi	the vector of individual outcomes and corresponding sampling variances.
fit.stats	a list with the log likelihood, deviance, AIC, and BIC values under the unrestricted and restricted likelihood.
...	some additional elements/values.

The results of the fitted model are neatly formatted and printed with the `print.rma.mh` function. If you also want the fit statistics, 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, and L'Abbe plots can be obtained with `forest.rma`, `funnel.rma`, `radial.rma`, and `labbe.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 assessor functions include `coef.rma`, `vcov.rma`, `logLik.rma`, `deviance.rma`, `AIC.rma`, and `BIC.rma`.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

Mantel, N. & Haenszel, W. (1959). Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the National Cancer Institute*, **22**, 719–748.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.uni](#), [rma.peto](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

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

rma.peto

Meta-Analysis via Peto's Method

Description

Function to fit a fixed-effects model for data from 2x2 tables via Peto's method.

Usage

```
rma.peto(ai, bi, ci, di, n1i, n2i,
         data, slab, subset,
         add=c(1/2,0), to=c("only0", "none"),
         level=95, digits=4)
```

Arguments

ai	vector to specify the 2x2 table frequencies (upper left cell). See below and the documentation of the escalc function for more details.
bi	vector to specify the 2x2 table frequencies (upper right cell). See below and the documentation of the escalc function for more details.
ci	vector to specify the 2x2 table frequencies (lower left cell). See below and the documentation of the escalc function for more details.
di	vector to specify the 2x2 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	an optional data frame containing the data supplied to the function.
slab	an optional vector with unique labels for the k studies.
subset	an optional vector indicating the subset of tables that should be used for the analysis. This can be a logical vector of length k or a numeric vector indicating the indices of the tables to include.
add	vector with two non-negative numbers indicating the amount to add to zero cells, counts, or frequencies. The first number is used in the calculation of the individual outcomes, the second number is used when applying Peto's method. See below and the documentation of the escalc function for more details.
to	a character vector with two strings indicating when the values under add should be added (either "all", "only0", "if0all", or "none"). The first string again applies when calculating the individual outcomes, the second string when applying Peto's method. See below and the documentation of the escalc function for more details.

level	a numerical value between 0 and 100 specifying the confidence interval level (default is 95).
digits	an integer specifying the number of decimal places to which the printed results should be rounded (default is 4).

Details

The studies are assumed to provide data in terms of 2x2 tables of the form:

	outcome 1	outcome 2	total
group 1	a _i	b _i	n _{1i}
group 2	c _i	d _i	n _{2i}

where a_i, b_i, c_i, and d_i denote the cell frequencies and n_{1i} and n_{2i} 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.

An approach for aggregating 2x2 table 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. Note that the printed results are given both in terms of the log and the raw units (for easier interpretation).

The method itself does not require the calculation of the individual (log) odds ratios and directly makes use of the 2x2 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 usually 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 individual (log) odds ratios for the k tables. 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 2x2 cell frequencies and under what circumstances when calculating the individual (log) odds ratios and when applying Peto's method. The documentation of the `escalc` function explains how the `add` and `to` arguments should be used. The first value of the `add` and `to` arguments applies when calculating the individual outcomes, the second value when applying Peto's method.

Value

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

<code>b</code>	aggregated log odds ratio.
<code>se</code>	standard error of the aggregated value.
<code>zval</code>	test statistics of the aggregated value.
<code>pval</code>	p-value for the test statistic.
<code>ci.lb</code>	lower bound of the confidence interval.
<code>ci.ub</code>	upper bound of the confidence interval.
<code>QE</code>	test statistic for the test of heterogeneity.

QEp	p-value for the test of heterogeneity.
k	number of tables 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, and BIC values under the unrestrained and restricted likelihood.
...	some additional elements/values.

The results of the fitted model are neatly formatted and printed with the `print.rma.peto` function. If you also want the fit statistics, 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, and L'abbe plots can be obtained with `forest.rma`, `funnel.rma`, `radial.rma`, and `labbe.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 assessor functions include `coef.rma`, `vcov.rma`, `logLik.rma`, `deviance.rma`, `AIC.rma`, and `BIC.rma`.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- 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**, 335–371.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.uni](#), [rma.mh](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### 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 the Linear (Mixed-Effects) Model***Description**

Function to fit the meta-analytic fixed- and random-effects models with or without moderators via the linear (mixed-effects) model. See the documentation of the [metafor-package](#) for more details on these models.

Usage

```
rma.uni(yi, vi, sei, ai, bi, ci, di, n1i, n2i,
        x1i, x2i, t1i, t2i, m1i, m2i, sd1i, sd2i,
        xi, mi, ri, ni, ti, mods, measure="GEN",
        intercept=TRUE, data, slab, subset,
        add=1/2, to="only0", vtype="LS",
        method="REML", weighted=TRUE, knha=FALSE,
        level=95, digits=4, btt, tau2, control)
rma(yi, vi, sei, ai, bi, ci, di, n1i, n2i,
    x1i, x2i, t1i, t2i, m1i, m2i, sd1i, sd2i,
    xi, mi, ri, ni, ti, mods, measure="GEN",
    intercept=TRUE, data, slab, subset,
    add=1/2, to="only0", vtype="LS",
    method="REML", weighted=TRUE, knha=FALSE,
    level=95, digits=4, btt, tau2, control)
```

Arguments

<code>yi</code>	vector of length k with the observed effect sizes or outcomes. See ‘Details’.
<code>vi</code>	vector of length k with the corresponding sampling variances. See ‘Details’.
<code>sei</code>	vector of length k with the corresponding standard errors. See ‘Details’.
<code>ai</code>	see below and the documentation of the escalc function for more details.
<code>bi</code>	see below and the documentation of the escalc function for more details.
<code>ci</code>	see below and the documentation of the escalc function for more details.
<code>di</code>	see below and the documentation of the escalc function for more details.
<code>n1i</code>	see below and the documentation of the escalc function for more details.
<code>n2i</code>	see below and the documentation of the escalc function for more details.
<code>x1i</code>	see below and the documentation of the escalc function for more details.
<code>x2i</code>	see below and the documentation of the escalc function for more details.
<code>t1i</code>	see below and the documentation of the escalc function for more details.
<code>t2i</code>	see below and the documentation of the escalc function for more details.
<code>m1i</code>	see below and the documentation of the escalc function for more details.
<code>m2i</code>	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.
ni	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.
mods	an 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 p' columns. Alternatively, a model formula can be used to specify the model. See 'Details'.
measure	a character string indicating the type of data supplied to the function. When <code>measure="GEN"</code> (default), the observed effect sizes or outcomes and corresponding sampling variances (or standard errors) should be supplied to the function via the <code>yi</code> , <code>vi</code> , and <code>sei</code> arguments (only one of the two, <code>vi</code> or <code>sei</code> , needs to be specified). Alternatively, one can set <code>measure</code> to one of the effect size or outcome measures described under the documentation for the escalc function and specify the needed data via the appropriate arguments.
intercept	logical, indicating whether an intercept term should be added to the model (default is TRUE).
data	an optional data frame containing the data supplied to the function.
slab	an optional vector with unique labels for the k studies.
subset	an optional vector indicating the subset of studies that should be used for the analysis. This can be a logical vector of length k or a numeric vector indicating the indices of the observations to include.
add	see the documentation of the escalc function.
to	see the documentation of the escalc function.
vtype	see the documentation of the escalc function.
method	a character string specifying whether a fixed- or a random/mixed-effects model should be fitted. A fixed-effects model (with or without moderators) is fitted when using <code>method="FE"</code> . Random/mixed-effects models are fitted by setting <code>method</code> equal to one of the following: "HE", "DL", "SJ", "ML", "REML", or "EB". Default is "REML". See 'Details'.
weighted	logical indicating whether weighted (default) or unweighted least squares should be used to fit the model.
knha	logical specifying whether the method by Knapp and Hartung (2003) should be used for adjusting test statistics and confidence intervals (default is FALSE). See 'Details'.
level	a numerical value between 0 and 100 specifying the confidence interval level (default is 95).
digits	an integer specifying the number of decimal places to which the printed results should be rounded (default is 4).

btt	an optional vector of indices specifying which coefficients to include in the omnibus test of moderators. See ‘Details’.
tau2	an optional numerical 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.
control	optional list of control values for the iterative estimation algorithms. Defaults to an empty list, which means that default values are defined inside the function. See ‘Note’.

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 odds ratios, log relative risks, 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 (or specify the standard errors, the square root of the sampling variances, via the *sei* argument). When specifying the data in this way, one must set *measure*="GEN" (which is the default).

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 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

Assuming the observed outcomes and corresponding sampling variances are supplied via *yi* and *vi*, the *fixed-effects model* is fitted with `rma(yi, vi, method="FE")`. The *random-effects model* is fitted with the same code but setting *method* to one of the various estimators for the amount of heterogeneity:

- `method="HS"` = Hunter-Schmidt estimator
- `method="HE"` = Hedges estimator
- `method="DL"` = DerSimonian-Laird estimator
- `method="SJ"` = Sidik-Jonkman estimator
- `method="ML"` = maximum-likelihood estimator
- `method="REML"` = restricted maximum-likelihood estimator
- `method="EB"` = empirical Bayes estimator.

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 design matrix with *k* rows and *p*’ columns (e.g., using `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 included in the model by default unless `intercept=FALSE`.

Alternatively, one can use the model `formula` syntax to specify the model. In this case, the `mods` argument should be set equal to a one-sided formula of the form `~ model` (e.g., `~ 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 term is controlled by the specified formula (e.g., `~ mod1 + mod2 + mod3 - 1` would exclude the intercept term).

In this case, the model formula determines whether an intercept is included in the model or not. A *fixed-effects with moderators model* is then fitted by setting `method="FE"`, while a *mixed-effects model* is fitted by specifying one of the estimators for the amount of (residual) heterogeneity given earlier.

Omnibus Test of Parameters

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, use `btt=c(3,4)` to only include the third and fourth coefficient from the model in the test (if an intercept is included in the model, then it corresponds to the first coefficient in the model).

Categorical Moderators

Categorical moderator variables can be included in the model in the same way that appropriately (dummy) coded categorical independent variables can be included in linear models. You can either do the dummy coding manually or use a model formula together with the `factor` function to let R handle the coding automatically. An example to illustrate these different approaches is provided below.

Knapp & Hartung Adjustment

By default, the test statistics of the individual coefficients in the model (and the corresponding confidence intervals) are based on the normal distribution, while the omnibus test is based on a chi-square distribution with m degrees of freedom (m being the number of coefficients tested). The Knapp and Hartung (2003) method (`knha=TRUE`) is an adjustment to the standard errors of the estimated coefficients, which helps to account for the uncertainty in the estimate of the amount of (residual) heterogeneity and leads to different reference distributions. Individual coefficients and confidence intervals are then based on the 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 (p being the total number of model coefficients including the intercept if it is present). The Knapp and Hartung (2003) adjustment is only meant to be used in the context of random- or mixed-effects models.

Value

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

<code>b</code>	estimated coefficients of the model.
<code>se</code>	standard errors of the coefficients.
<code>zval</code>	test statistics of the coefficients.
<code>pval</code>	p-values for the test statistics.
<code>ci.lb</code>	lower bound of the confidence intervals for the coefficients.
<code>ci.ub</code>	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	estimated standard error of the estimated amount of (residual) heterogeneity when using ML or REML estimation (NA otherwise).
k	number of outcomes included in the model fitting (equal to length(yi) unless subset was used or if there are missing data).
p	number of coefficients in the model (including the intercept).
m	number of coefficients included in the omnibus test of coefficients.
QE	test statistic for the test of (residual) heterogeneity.
QEp	p-value for the test of (residual) heterogeneity.
QM	test statistic for the omnibus test of coefficients.
QMp	p-value for the omnibus test of coefficients.
I2	value of I^2 (only for the random-effects model; NA otherwise).
H2	value of H^2 (only for the random-effects model; NA otherwise).
int.only	logical that indicates whether the model only includes an intercept.
yi, vi, X	the vector of outcomes, the corresponding sampling variances, and the design matrix of the model.
fit.stats	a list with the log likelihood, deviance, AIC, and BIC values under the unrestricted and restricted likelihood.
...	some additional elements/values.

The results of the fitted model are neatly formatted and printed with the `print.rma.uni` function. If you also want the fit statistics, 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.uni`. Permutation tests for the model coefficient(s) can be obtained with `permutest.rma.uni`.

Predicted/fitted values can be obtained with `predict.rma` and `fitted.rma`. For best linear unbiased predictions, see `blup.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`.

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, and L'abbe plots (the latter two only for models without moderators) can be obtained with `forest.rma`, `funnel.rma`, `radial.rma`, and `labbe.rma`. 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.

Tests for publication bias (or more accurately, for funnel plot asymmetry) 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.

For models without moderators, a cumulative meta-analysis (i.e., adding one observation at a time) can be obtained with `cumul.rma.uni`.

Other assessor 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, HE, DL, and SJ estimators of τ^2 are based on closed-form solutions, the ML, REML, and EB estimators must be obtained numerically. For this, the `rma()` 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 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). 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)` when calling the `rma()` function. The step length of the Fisher scoring algorithm can also be manually adjusted by a desired factor with `control=list(stepadj=value)` (values below 1 will reduce the step length). Information on the evolution of the algorithm is obtained with `control=list(verbose=TRUE)`.

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, HE, and DL estimators, negative estimates are therefore truncated to zero. For ML, REML, and EB estimation, the Fisher scoring algorithm makes use of step halving to guarantee a non-negative estimate. For those brave enough to step into risky territory, there is the option to set the lower bound of τ^2 equal to some other value besides zero with `control=list(tau2.min=value)`.

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 the random-effects model not specific to any particular outcome measure is described in Viechtbauer (2005). The same idea can be easily extended to the mixed-effects model and is implemented here.

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 turns out to be zero.

If you get a ‘singular matrix’ error when using the function, then this means that there is a linear relationship between the moderator variables included in the model. For example, two moderators that correlated perfectly would cause this error. 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 are warranted:

- The sampling variances (i.e., the v_i values) are treated as if they were 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 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 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>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- DerSimonian, R. & Laird, N. (1986). Meta-analysis in clinical trials. *Controlled Clinical Trials*, **7**, 177–188.
- Harville, D. A. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, **72**, 320–338.
- Hedges, L. V. (1983). A random effects model for effect sizes. *Psychological Bulletin*, **93**, 388–395.
- Hedges, L. V. & Olkin, I. (1985). *Statistical methods for meta-analysis*. San Diego, CA: Academic Press.
- Hunter, J. E. & Schmidt, F. L. (1990). *Methods of meta-analysis: Correcting error and bias in research findings*. Newbury Park, CA: Sage.
- Jennrich, R. I. & Sampson, P. F. (1976). Newton-Raphson and related algorithms for maximum likelihood variance component estimation. *Technometrics*, **18**, 11–17.
- Knapp, G. & Hartung, J. (2003). Improved tests for a random effects meta-regression with a single covariate. *Statistics in Medicine*, **22**, 2693–2710.
- Morris, C. N. (1983). Parametric Empirical Bayes inference: Theory and applications (with discussion). *Journal of the American Statistical Association*, **78**, 47–65.
- Raudenbush, S. W. (2009). Analyzing effect sizes: Random effects models. In H. C. 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**, 823–838.
- Sidik, K. & Jonkman, J. N. (2005b). Simple heterogeneity variance estimation for meta-analysis. *Journal of the Royal Statistical Society, Series C*, **54**, 367–384.
- Viechtbauer, W. (2005). Bias and efficiency of meta-analytic variance estimators in the random-effects model. *Journal of Educational and Behavioral Statistics*, **30**, 261–293.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.mh](#), [rma.peto](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

### random-effects model
rma(yi, vi, data=dat, method="REML")

### mixed-effects model with two moderators (absolute latitude and publication year)
rma(yi, vi, mods=cbind(ablat, year), data=dat, method="REML")

### using a model formula to specify the same model
rma(yi, vi, mods=~ablat+year, data=dat, method="REML")

### supplying the cell frequencies directly to the function
rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat, year),
    data=dat, measure="RR", method="REML")

### 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
### note: the intercept is the first coefficient, so btt=c(2,3)
rma(yi, vi, mods=cbind(alloc.random, alloc.systematic, year, ablat),
    data=dat, method="REML", btt=c(2,3))

### using a model formula to specify the same model
rma(yi, vi, mods=~factor(alloc)+year+ablat, data=dat, method="REML", btt=c(2,3))

### 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 get 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
res <- rma(yi, vi, mods=cbind(ablat, year), data=dat, method="FE")
res
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$QE ### Q-value within subgroup
res.r$QE ### Q-value within subgroup
res.s$QE ### Q-value within subgroup

res$QE
res.a$QE + res.r$QE + res.s$QE

```

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.ztor.int(xi, targs, ...)
transf.exp.int(xi, targs, ...)
transf.ilogit.int(xi, targs, ...)

```

Arguments

<code>xi</code>	a vector of values to be transformed.
<code>ni</code>	a vector of sample sizes.
<code>ti</code>	a vector of person-times at risk.
<code>targs</code>	a list with additional arguments for the transformation function. See ‘Details’.
<code>...</code>	other arguments.

Details

The following transformation functions are currently implemented:

- `transf.rtoz`: Fisher’s r-to-z transformation.
- `transf.ztor`: inverse of the Fisher’s r-to-z transformation.
- `transf.logit`: logit (log odds) transformation.
- `transf.ilogit`: inverse of the logit transformation.
- `transf.arcsin`: arcsine transformation.
- `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-time at risk.
- `transf.iirft`: inverse of the Freeman-Tukey transformation for incidence rates.
- `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.

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.

Value

A vector with the transformed values.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

References

- Fisher, R. A. (1921). On the “probable error” of a coefficient of correlation deduced from a small sample. *Metron*, **1**, 1–32.
- Freeman, M. F. & Tukey, J. W. (1950). Transformations related to the angular and the square root. *Annals of Mathematical Statistics*, **21**, 607–611.
- Miller, J. J. (1978). The inverse of the Freeman-Tukey double arcsine transformation. *American Statistician*, **32**, 138.
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

Examples

```
### load BCG vaccine data
data(dat.bcg)

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

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

### average relative risk 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 Method

Description

The function trimfill is generic. It can be used to apply the trim and fill method proposed by Duval and Tweedie (2000a, 2000b; see also Duval, 2005) to specific classes of objects.

Usage

```
trimfill(x, ...)
```

Arguments

x an object of class "rma.uni".
... other arguments.

Details

Currently, there is only a method for objects of class "rma.uni" created by the `rma.uni` function. Accordingly, the corresponding method is called `trimfill.rma.uni`. See the documentation for that function for more details.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

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**, 455–463.
- 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**, 89–98.
- Duval, S. J. (2005). The trim and fill method. In H. R. Rothstein, A. J. Sutton, & M. Borenstein (Eds.) (2005) *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. <http://www.jstatsoft.org/v36/i03/>.

See Also

[trimfill.rma.uni](#)

trimfill.rma.uni *Trim and Fill Method for rma.uni Objects*

Description

Apply the trim and fill method to objects of class "rma.uni".

Usage

```
## S3 method for class 'rma.uni'  
trimfill(x, side, estimator="LO", maxit=50, verbose=FALSE, ...)
```

Arguments

x	an object of class "rma.uni".
side	either "left" or "right", indicating on which side of the funnel plot the missing studies should be imputed. If left undefined, the side is chosen within the function depending on the results of Egger's regression test (see regtest for details on this test).
estimator	either "L0" or "R0" indicating the estimator to use for estimating the number of missing studies.
maxit	an integer value indicating the maximum number of iterations to use for the trim and fill method.
verbose	logical indicating whether information about the evolution of the algorithm should be printed.
...	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 the 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. The trim and fill method can only be used in the context of the 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.
fill	a vector of dummy codes, indicating which of the data are the observed (0) and the augmented (1) 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.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
 project homepage: <http://www.metafor-project.org/>
 author homepage: <http://www.wvbauer.com/>

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**, 455–463.
- 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**, 89–98.
- Duval, S. J. (2005). The trim and fill method. In H. R. Rothstein, A. J. Sutton, & M. Borenstein (Eds.) (2005) *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. <http://www.jstatsoft.org/v36/i03/>.

See Also

[funnel.rma](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a fixed-effects model
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg,
          data=dat.bcg, measure="RR", method="FE")
res.tf <- trimfill(res)
res.tf
funnel(res.tf)

### meta-analysis of the log relative risks using a random-effects model
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg,
          data=dat.bcg, measure="RR", method="REML")
res.tf <- trimfill(res)
res.tf
funnel(res.tf)
```

vcov.rma

Variance-Covariance Matrix of Parameter Estimates for rma Objects

Description

The function extracts the (estimated) variance-covariance matrix of the fixed effects parameter estimates from objects of class "rma".

Usage

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

Arguments

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

Value

A matrix corresponding to the variance-covariance matrix.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.uni](#), [rma.mh](#), [rma.peto](#)

Examples

```
### load BCG vaccine data
data(dat.bcg)

### meta-analysis of the log relative risks using a mixed-effects model
### with two moderators (absolute latitude and publication year)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat, year),
          data=dat.bcg, measure="RR", method="REML")
vcov(res)
```

weights.rma.uni

Weights for rma Objects

Description

The function extracts the weights (in %) given to the observed effects or outcomes during the model fitting from objects of class "rma.uni", "rma.mh", and "rma.peto".

Usage

```
## S3 method for class 'rma.uni'  
weights(object, ...)  
## S3 method for class 'rma.mh'  
weights(object, ...)  
## S3 method for class 'rma.peto'  
weights(object, ...)
```

Arguments

object an object of class "rma.uni", "rma.mh", or "rma.peto".
... other arguments.

Value

A vector with the weights.

Author(s)

Wolfgang Viechtbauer <wvb@metafor-project.org>
project homepage: <http://www.metafor-project.org/>
author homepage: <http://www.wvbauer.com/>

References

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36**(3), 1–48. <http://www.jstatsoft.org/v36/i03/>.

See Also

[rma.uni](#), [rma.mh](#), [rma.peto](#), [influence.rma.uni](#)

Examples

```
### load BCG vaccine data  
data(dat.bcg)  
  
### meta-analysis of the log relative risks using a mixed-effects model  
### with two moderators (absolute latitude and publication year)  
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods=cbind(ablat, year),  
          data=dat.bcg, measure="RR", method="REML")  
weights(res)
```

Index

*Topic **IO**

metafor.news, 75

*Topic **aplot**

addpoly.default, 7

addpoly.rma, 10

*Topic **datagen**

escalc, 35

*Topic **datasets**

dat.bcg, 27

dat.co2, 29

dat.empint, 31

dat.los, 32

dat.warfarin, 34

*Topic **hplot**

forest, 47

forest.cumul.rma, 48

forest.default, 51

forest.rma, 54

funnel, 61

funnel.rma, 62

labbe, 68

labbe.rma, 69

plot.infl.rma.uni, 80

plot.rma.uni, 82

qqnorm.rma.uni, 95

radial, 97

radial.rma, 98

*Topic **htest**

fsn, 59

ranktest, 100

ranktest.rma, 101

regtest, 103

regtest.rma, 104

*Topic **manip**

transf, 124

*Topic **methods**

addpoly, 6

blup, 14

cumul, 22

cumul.rma.mh, 23

cumul.rma.uni, 25

fitstats, 43

leave1out, 71

leave1out.rma.mh, 72

leave1out.rma.uni, 74

permutest, 76

trimfill, 126

*Topic **models**

anova.rma.uni, 12

blup.rma.uni, 15

coef.permutest.rma.uni, 17

coef.rma, 18

confint.rma.uni, 20

fitstats.rma, 44

fitted.rma, 46

hatvalues.rma.uni, 65

influence.rma.uni, 66

permutest.rma.uni, 77

predict.rma, 83

residuals.rma, 107

rma.mh, 109

rma.peto, 113

rma.uni, 116

trimfill.rma.uni, 127

vcov.rma, 129

weights.rma.uni, 130

*Topic **package**

metafor-package, 3

*Topic **print**

print.anova.rma.uni, 86

print.fsn, 87

print.infl.rma.uni, 88

print.list.rma, 89

print.permutest.rma.uni, 90

print.predict.rma, 91

print.ranktest.rma, 92

print.regtest.rma, 93

print.rma.uni, 94

- addpoly, 6, 53, 54, 57, 58
- addpoly.default, 7, 7
- addpoly.rma, 7, 10
- AIC, 44
- AIC.rma, 112, 115, 121
- AIC.rma (fitstats.rma), 44
- anova.rma.uni, 12, 45, 87, 120

- BIC.rma, 112, 115, 121
- BIC.rma (fitstats.rma), 44
- blup, 14
- blup.rma.uni, 14, 15, 46, 47, 85, 120

- coef.permutest.rma.uni, 17, 79
- coef.rma, 18, 112, 115, 121
- confint, 20
- confint.rma.mh (confint.rma.uni), 20
- confint.rma.peto (confint.rma.uni), 20
- confint.rma.uni, 20, 120
- cor.test, 102
- cumul, 22, 24, 26, 50
- cumul.rma.mh, 23, 23, 47, 112
- cumul.rma.peto, 23, 47, 115
- cumul.rma.peto (cumul.rma.mh), 23
- cumul.rma.uni, 23, 25, 47, 120

- dat.bcg, 27, 38
- dat.co2, 29, 39
- dat.empint, 31, 40
- dat.los, 32, 39
- dat.warfarin, 34, 39
- deviance.rma, 112, 115, 121
- deviance.rma (fitstats.rma), 44

- escalc, 3, 4, 6, 35, 70, 109–111, 113, 114, 116–118

- factor, 119
- file.show, 76
- fitstats, 43
- fitstats.rma, 43, 44, 44, 112, 115, 120
- fitted.rma, 16, 46, 85, 120
- forest, 8, 11, 47, 50, 54, 58, 83
- forest.cumul.rma, 24, 26, 47, 48, 48
- forest.default, 7, 9, 11, 47, 48, 51
- forest.rma, 7, 9, 11, 47, 48, 54, 112, 115, 120
- formula, 117, 119
- fsn, 59, 88
- funnel, 61, 83
- funnel.rma, 61, 62, 62, 112, 115, 120, 128, 129

- galbraith (radial), 97

- hatvalues.rma.uni, 65, 68, 121

- influence.rma.uni, 64, 65, 66, 75, 82, 88, 89, 108, 109, 120, 131

- labbe, 68
- labbe.rma, 69, 69, 112, 115, 120
- leave1out, 71, 73, 75
- leave1out.rma.mh, 71, 72, 72, 112
- leave1out.rma.peto, 71, 72, 115
- leave1out.rma.peto (leave1out.rma.mh), 72
- leave1out.rma.uni, 71, 72, 74, 120
- logLik.rma, 112, 115, 121
- logLik.rma (fitstats.rma), 44

- metafor (metafor-package), 3
- metafor-package, 3, 75, 116
- metafor.news, 5, 75

- par, 80
- permutest, 76
- permutest.rma.uni, 18, 76, 77, 77, 91, 120
- plot.infl.rma.uni, 67, 68, 80
- plot.rma.mh, 112
- plot.rma.mh (plot.rma.uni), 82
- plot.rma.peto, 115
- plot.rma.peto (plot.rma.uni), 82
- plot.rma.uni, 82, 120
- points, 49, 53, 56, 63, 70, 80, 96, 99
- predict.rma, 16, 46, 47, 83, 91, 120
- print.anova.rma.uni, 13, 86
- print.fsn, 60, 87
- print.infl.rma.uni, 67, 68, 88
- print.list.rma, 15, 24, 26, 73, 74, 89, 108
- print.permutest.rma.uni, 79, 90
- print.predict.rma, 85, 91
- print.ranktest.rma, 92, 102
- print.regtest.rma, 93, 105
- print.rma.mh, 112
- print.rma.mh (print.rma.uni), 94
- print.rma.peto, 115
- print.rma.peto (print.rma.uni), 94
- print.rma.uni, 94, 120, 128

qqnorm.rma.mh, 112
qqnorm.rma.mh (qqnorm.rma.uni), 95
qqnorm.rma.peto, 115
qqnorm.rma.peto (qqnorm.rma.uni), 95
qqnorm.rma.uni, 83, 95, 120

radial, 83, 97
radial.rma, 97, 98, 98, 112, 115, 120
ranktest, 61, 100, 102, 106
ranktest.rma, 92, 101, 101, 120
regtest, 102, 103, 106, 128
regtest.rma, 93, 103, 104, 104, 120
residuals.rma, 64, 107, 112, 115, 120
rma, 3
rma (rma.uni), 116
rma.mh, 5–7, 19, 42, 45, 47, 64, 70, 71, 95, 97,
100, 107, 109, 109, 115, 123, 130,
131
rma.peto, 5–7, 19, 42, 45, 47, 64, 70, 71, 95,
97, 100, 107, 109, 112, 113, 123,
130, 131
rma.uni, 3, 6, 7, 13, 14, 16–20, 22, 42, 45, 47,
64, 65, 70, 71, 76, 79, 85, 95, 97,
100, 107–109, 112, 115, 116, 127,
128, 130, 131
rstandard.rma.mh, 112
rstandard.rma.mh (residuals.rma), 107
rstandard.rma.peto, 115
rstandard.rma.peto (residuals.rma), 107
rstandard.rma.uni, 120
rstandard.rma.uni (residuals.rma), 107
rstudent.rma.mh, 112
rstudent.rma.mh (residuals.rma), 107
rstudent.rma.peto, 115
rstudent.rma.peto (residuals.rma), 107
rstudent.rma.uni, 67, 68, 120
rstudent.rma.uni (residuals.rma), 107

summary.rma, 112, 115, 120
summary.rma (print.rma.uni), 94
supsmu, 96

transf, 124
trimfill, 61, 63, 64, 126
trimfill.rma.uni, 120, 127, 127

uniroot, 21

vcov.rma, 112, 115, 121, 129

weights.rma.mh (weights.rma.uni), 130
weights.rma.peto (weights.rma.uni), 130
weights.rma.uni, 121, 130