Package ‘metafor’

October 8, 2009

Version 0.5-5

Date 2009-10-08

Title Meta-Analysis Package for R

Author Wolfgang Viechtbauer <wvb@wvbauer.com>

Maintainer Wolfgang Viechtbauer <wvb@wvbauer.com>

Depends R (>= 2.9.0), stats

Description The metafor package consists of a collection of functions for conducting meta-analyses in R. Fixed- and random-effects models (with and without moderators) can be fitted via the general linear (mixed-effects) model. For 2x2 table data, the Mantel-Haenszel and Peto’s method are also implemented.

License GPL (>= 2)

URL http://www.wvbauer.com/

Repository CRAN

Date/Publication 2009-10-08 12:41:35

R topics documented:

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

Description

The **metafor** package provides functions for conducting meta-analyses in R. Currently, there are functions to fit the meta-analytic fixed- and random-effects model via the general linear (mixed-effects) model, the Mantel-Haenszel method, and Peto's method (the latter two only for fixed-effects models). Moderators (study-level covariates) can be included when using the general linear (mixed-effects) model approach, allowing the user to fit 'meta-regression' models. The package also provides various plot functions (for example, for forest, funnel, and radial plots) and functions for assessing the model fit and obtaining case diagnostics.

The rma.uni Function

The various meta-analytic models that are usually applied in practice are special cases of the general linear (mixed-effects) model. The `rma.uni` function (with alias `rma`) provides a general framework for fitting the various 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 risk ratios, 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, please see the documentation of the `escalc` function. The notation and models underlying the `rma.uni` function are explained below.

For a set of $i = 1, \ldots, k$ 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 (unknown) true effects or outcomes with sampling variances equal to v_i. The v_i values are assumed to be known.

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 $\sum_{i=1}^k w_i \theta_i / \sum_{i=1}^k w_i$, that is, the *weighted average* of the true effects in the set of k studies, with weights equal to $w_i = 1/v_i$. 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 $1/k \sum_{i=1}^k \theta_i$).

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. The true effects or outcomes in this 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 population of studies). 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 in the true effects or outcomes over and beyond the heterogeneity that is introduced by the moderators).

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). Therefore, the fixed-effects model may be used even when the true effects or outcomes are heterogeneous. However, when using a fixed-effects model, we can only draw conclusions about the average effect (weighted or unweighted) in the set of 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.

Moreover, the fixed-effects with moderators model does *not* assume that the heterogeneity in the true effects or outcomes is fully accounted for by the moderators included in the model. However, when using a fixed-effects with moderators model, we must again restrict our inferences to the set of k studies included in the meta-analysis. For more details on the distinction between fixed- and random-effects models, see Hedges and Vevea (1998) and Laird and Mosteller (1990).

Note that the observed effects or outcomes are assumed to be independent in the `rma.uni` function. Functions to handle multivariate situations and correlated outcomes will be included in the package at a later point.

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 (Mantel & Haenszel, 1959). 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). The Mantel-Haenszel method is implemented in the `rma.mh` function. It can be used in combination with odds ratios, risk ratios, and risk differences. 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.

Author(s)

Wolfgang Viechtbauer
e-mail: ⟨wvb@www.wvbauer.com⟩
web: 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, include the package name (i.e., 'metafor') in your post.

References

Cooper, H. C. & Hedges, L. V. (1994) *The handbook of research synthesis*. 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.

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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

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, row=-1, level=95, digits=2, annotate=TRUE,
        mlab=NULL, transf=FALSE, atransf=FALSE, targs=NULL,
        col="black", efac=1, cex=NULL, ...)
```

Arguments

`x`	a vector with the values at which the polygons should be drawn.
`vi`	a vector with the corresponding variances.
`sei`	a vector with the corresponding standard errors. (note: only one of the two, `vi` or `sei`, needs to be specified)
`row`	integer specifying the (starting) row of where the polygon(s) should be placed (default is -1).
`level`	a numerical value between 0 and 100 specifying the confidence interval level (default is 95).
`digits`	integer value specifying the number of decimal places to which the annotations should be rounded (default is 2).

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

`mlab` optional character vector with the same length as `x` giving labels for the polygons that are drawn. Defaults to `NULL`, which suppresses labels.

`transf` an optional argument specifying the name of a function that should be used to transform the `x` values and confidence interval bounds. 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 polygons that are drawn (default is `"black"`).

`efac` vertical expansion factor for the polygons. 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 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.

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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

See Also

`forest.rma`, `forest.default`

Examples

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

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

### forest plot of the observed risk ratios
forest(res, addfit=FALSE, atransf=exp, addrows=3, xlab="Relative Risk (log scale)")

### predicted log average risk ratios for 10, 30, and 50 degrees absolute latitude
```

```
x <- predict(res, newmods=c(10, 30, 50))

### add predicted average risk ratios to forest plot
addpoly(x$pred, sei=x$se, atransf=exp,
        mlab=c("10 Degrees", "30 Degrees", "50 Degrees"))
```

addpoly.rma *Add Summary Estimate Polygon to Forest Plot*

Description

Function to add a polygon to a forest plot showing the summary estimate with correspondong 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, atransf=FALSE, targs=NULL,
         col="black", efac=1, cex=NULL, ...)
```

Arguments

`x` an object of class "rma".

`row` integer specifying the row number where the polygon should be placed (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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

See Also

`forest.rma`, `forest.default`

Examples

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

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

### forest plot of the observer risk ratios with summary estimate
forest(res, atransf=exp, addrows=1, xlab="Relative Risk (log scale)")

### meta-analysis of the log risk ratios 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, atransf=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 formated 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@www.wvbauer.com⟩; 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.

See Also

rma.uni

Examples

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

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

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

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

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@www.wvbauer.com⟩; 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

---

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

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=NULL, ...)
```

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 formated 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.uni`.

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.uni`.

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@www.wvbauer.com⟩; `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.

See Also

rma.uni, predict.rma.uni, fitted.rma

Examples

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

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

### meta-analysis of the log risk ratios 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)
```

---

cint — *Confidence Intervals for Model Parameters*

Description

The function `cint` is generic. It provides confidence intervals for specific classes of objects.

Usage

```
cint(object, ...)
```

Arguments

`object`	an object for which confidence intervals can be calculated. 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 `cint.rma.uni`. See the documentation for that function for more details.

Value

Confidence interval bounds of the model parameters.

Author(s)

Wolfgang Viechtbauer; ⟨wvb@www.wvbauer.com⟩; `http://www.wvbauer.com/`

See Also

`cint.rma.uni`

`cint.rma.uni` *Confidence Intervals for rma.uni Objects*

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':
cint(object, fixed=FALSE, random=TRUE, level=object$level,
     digits=object$digits, control=list(), ...)
```

Arguments

`object`	an object of class `"rma.uni"`.
`fixed`	logical indicating whether confidence intervals for the model coefficients should be returned (default is `FALSE`).
`random`	logical indicating whether a confidence interval for the amount of (residual) heterogeneity should be returned (default is `TRUE`).
`level`	a numerical value between 0 and 100 specifying the confidence interval level (the default is to take the value from the object).
`digits`	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).
`control`	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’.
`...`	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 or when calling `coef.rma`.

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:

`estimate` estimate of the model coefficient or variance component

`ci.lb` lower bound of the confidence interval.

`ci.ub` 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, uing the empirical Bayes estimator of τ^2 when fitting the model guarantees that the estimate of τ^2 falls within the confidence interval (as long as `level` is larger than 26).

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@www.wvbauer.com⟩; 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.

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

See Also

`rma.uni`

Examples

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

### meta-analysis of the log risk ratios 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
cint(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
cint(res)
```

`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@www.wvbauer.com⟩; http://www.wvbauer.com/

See Also

rma.uni, rma.mh, rma.peto

Examples

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

### meta-analysis of the log risk ratios 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)
```

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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

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=NULL, digits=x$digits, transf=FALSE, ...)
## S3 method for class 'rma.peto':
cumul(x, order=NULL, digits=x$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 risk ratios (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 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 object is formated 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@www.wvbauer.com⟩; http://www.wvbauer.com/

See Also

cumul, forest.cumul.rma

Examples

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

### meta-analysis of the (log) risk ratios 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)

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=NULL, digits=x$digits, transf=FALSE, targs=NULL, ...)
```

Arguments

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

`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` an optional argument specifying the name of a function that should be used to transform the model coefficients 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

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:

`estimate` estimated coefficients of the model.

`se` standard errors of the coefficients. `NA` if `transf` is used to transform the coefficients.

`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.
`tau2`	estimated amounts of (residual) heterogeneity (only for random-effects models).
`I2`	values of I^2 (only for random-effects models).
`H2`	values of H^2 (only for random-effects models).

The object is formated 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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

See Also

`cumul`, `forest.cumul.rma`

Examples

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

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

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

cumul(res, transf=exp, order=order(dat$year))
```

`dat.bcg` *Data for BCG Vaccine Studies*

Description

Results from 13 clinical trials examining the effectivess 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 where the study was conducted
alloc	`character`	method of treatment allocation (random, alternate, or systematic)

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

Source

Colditz, G. A., Brewer, T. F., Berkey, C. S., Wilson, M. E., Burdick, E., Fineberg, H. V. & Mosteller, F. (1994) Efficacy of BCG vaccine in the prevention of tuberculosis: Meta-analysis of the published literature. *Journal of the American Medical Association*, **271**, 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.

Examples

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

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

```
res

### average risk ratio 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=ablat, measure="RR", data=dat.bcg, method="DL")
res

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

escalc *Calculate Effect Size and Outcome Measures*

Description

The function calculates various effect size and outcome measures that are commonly used in meta-analyses. The corresponding sampling variances are also calculated.

Usage

```
escalc(measure, ai, bi, ci, di, n1i, n2i, m1i, m2i, sd1i, sd2i,
       xi, mi, ri, ni, data=NULL, 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 what arguments below should then be specified.

`ai` vector to specify 2x2 table frequencies (upper left cell).

`bi` vector to specify 2x2 table frequencies (upper right cell).

`ci` vector to specify 2x2 table frequencies (lower left cell).

`di` vector to specify 2x2 table frequencies (lower right cell).

`n1i` vector to specify group sizes or row total (first group/row).

`n2i` vector to specify group sizes or row total (second group/row).

`m1i` vector to specify means (first group).

`m2i` vector to specify means (second group).

`sd1i` vector to specify standard deviations (first group).

`sd2i` vector to specify standard deviations (second group).

`xi` vector to specify frequencies of the event of interest.

`mi` vector to specify frequencies of the complement of the event of interest.

`ri` vector to specify the raw correlation coefficients.

`ni`	vector to specify the sample sizes.
`data`	an optional data frame containing the variables given to the arguments above.
`add`	See 'Details'.
`to`	See 'Details'.
`vtype`	See 'Details'.

Details

Raw and Standardized Mean Difference

The raw mean difference and standardized mean difference are useful effect size measure when meta-analyzing a set of studies comparing two groups (e.g., treatment and control groups) with respect to some quantitative (and ideally normally distributed) dependent variable. For these outcome measures, `m1i` and `m2i` denote the means of the two groups, `sd1i` and `sd2i` the standard deviations of the scores in the two groups, and `n1i` and `n2i` the sample sizes of the two groups.

- `"MD"`: The *raw mean difference* is equal to `m1i-m2i`.
- `"SMD"`: The *standardized mean difference* is equal to `(m1i-m2i)/spi`, where `spi` is the pooled standard deviation of the two groups (which is calculated inside of the function). 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"`.

Effect Size and Outcome Measures for 2x2 Table Data

Studies providing data in terms of 2x2 tables call for other outcome measures. In particular, assume that we have tables of the form:

	outcome 1	outcome 2	total
group 1	`ai`	`bi`	`n1i`
group 2	`ci`	`di`	`n2i`

where `ai`, `bi`, `ci`, and `di` denote the cell frequencies and `n1i` and `n2i` the row totals. For example, in a set of RCTs, group 1 and group 2 may refer to the treatment and placebo group, with outcome 1 denoting some event of interest 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) 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 risk ratio (also called relative risk), the risk difference, and the arc-sine transformed risk difference. The phi coefficient, Yule's Q, and Yule's Y are additional measures of association for 2x2 table data (but they may not be the most ideal choices for meta-analyses of such data). For these measures, one needs to supply either `ai`, `bi`, `ci`, and `di` or alternatively `ai`, `ci`, `n1i`, and `n2i`. Note that the log is taken of the risk and the odds ratio, which makes these outcome measures symmetric about 0 and helps to make the distribution of these outcome measure closer to normal.

- `"RR"`: The *log relative risk* is equal to the log of `(ai/n1i)/(ci/n2i)`.
- `"OR"`: The *log odds ratio* is equal to the log of `(ai*di)/(bi*di)`.
- `"RD"`: The *risk difference* is equal to `(ai/n1i)-(ci/n2i)`.
- `"AS"`: The *arc-sine transformed risk difference* is equal to `asin(sqrt(ai/n1i)) - asin(sqrt(ci/n2i))`. 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 `(ai-si*n1i/ni)/((si*ti*n1i*n2i)/(ni^2*(ni-1)))`, where `si=ai+ci`, `ti=bi+di`, and `ni=n1i+n2i`. Note that this measure technically assumes that the true odds ratio is equal to 1 in all tables.
- `"PHI"`: The *phi coefficient* is equal to `(ai*di-bi*ci)/sqrt(n1i*n2i*si*ti)`, where `si=ai+ci` and `ti=bi+di`.
- `"YUQ"`: *Yule's Q* is equal to `(oi-1)/(oi+1)`, where `oi` is the odds ratio.
- `"YUY"`: *Yule's Y* is equal to `(sqrt(oi)-1)/(sqrt(oi)+1)`, where `oi` is the odds ratio.

Cell entries with a zero can be problematic, especially for the risk and the odds ratio. Adding a 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 the 2x2 tables in all k tables. When `to="only0"`, the value of `add` is added to each cell of the 2x2 tables only in those tables with at least one cell equal to 0. When `to="if0all"`, the value of `add` is added to each cell of the 2x2 tables in all k tables, but only when there is at least one 2x2 table with a zero entry. 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`).

Proportions and Transformations Thereof

When the studies provide data for a single group with respect to a dichotomous dependent variable, then the raw proportion, log transformed proportion, logit transformed proportion (i.e., log odds), the arc-sine transformed proportion, and the Freeman-Tukey double arc-sine transformed proportion are useful outcome measures. Here, one needs to specify `xi` and `ni`, denoting the number of individuals experiencing the event of interest and the total number of individuals, respectively. Instead of specifying `ni`, one can use `mi` to specify the number of individuals that do not experience the event of interest.

- `"PR"`: The *raw proportion* is equal to `xi/ni`.
- `"PLN"`: The *log transformed proportion* is equal to the log of `xi/ni`.
- `"PLO"`: The *logit transformed proportion* is equal to the log of `xi/(ni-xi)`.
- `"PAS"`: The arc-sine transformation is a variance stabilizing transformation for proportions. The *arc-sine transformed proportion* is equal to `asin(sqrt(xi/ni))`.
- `"PFT"`: Yet another variance stabilizing transformation for proportions was suggested by Freeman & Tukey (1950). The *Freeman-Tukey double arc-sine transformed proportion* is equal to `1/2*(asin(sqrt(xi/(ni+1))) + asin(sqrt((xi+1)/(ni+1))))`.

Again, zero cell entries can be problematic. When `to="all"`, the value of `add` is added to `xi` and `mi` in all k studies. When `to="only0"`, the value of `add` is added only for studies where the `xi` or `mi` 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 `xi` or `mi`. Setting `to="none"` or `add=0` again means that no adjustment to the observed values is made.

Raw and Transformed Correlation Coefficient

Another frequently used outcome measure in meta-analyses is the correlation coefficient. Here, one needs to specify `ri`, the vector with the raw correlation coefficients, and `ni`, the corresponding sample sizes.

- `"COR"`: The *raw correlation coefficient* is simply equal to `ri` 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 approximations or the 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+ri)/(1-ri))`.

Value

A data frame with the following elements:

`yi`	value of the effect size or outcome measure.
`vi`	corresponding (estimated) sampling variance.

Author(s)

Wolfgang Viechtbauer; ⟨wvb@www.wvbauer.com⟩; `http://www.wvbauer.com/`

References

Cooper, H. C. & Hedges, L. V. (1994) *The handbook of research synthesis*. 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.

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.

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

### add log risk ratios and sampling variances to the data frame
dat <- cbind(dat.bcg, dat)
dat
```

---

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(x, ...)
```

Arguments

`x`	an object for which fit statistics and information criteria can be calculated. See 'Details'.
`...`	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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

See Also

`fitstats.rma`

`fitstats.rma` *Fit Statistics and Information Criteria for rma Objects*

Description

The function extracts the log likelihood, deviance, AIC, and BIC values from objects of class `"rma"`.

Usage

```
## S3 method for class 'rma':
fitstats(x, REML=NULL, ...)
```

Arguments

`x`	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. When `NULL` (default), the regular likelihood is used, unless restricted maximum likelihood estimation was used to fit the model.
`...`	other arguments.

Value

A column vector with the log likelihood, deviance, AIC, and BIC values.

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.

Author(s)

Wolfgang Viechtbauer; ⟨wvb@www.wvbauer.com⟩; `http://www.wvbauer.com/`

See Also

`rma.uni`, `rma.mh`, `rma.peto`

Examples

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

### meta-analysis of the log risk ratios 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)
```

---

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

For rma.uni objects, the predict.rma.uni 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.

For objects not involving moderators, the fitted values are all identical to the estimated value of the model intercept.

Author(s)

Wolfgang Viechtbauer; ⟨wvb@www.wvbauer.com⟩; http://www.wvbauer.com/

See Also

predict.rma.uni, blup.rma.uni

Examples

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

### meta-analysis of the log risk ratios 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@www.wvbauer.com⟩; http://www.wvbauer.com/

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,
       atransf=FALSE, targs=NULL, addrows=0, 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 and corresponding labels 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 and 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 (NULL by default, which suppresses the title).

`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.

`ilab.pos` integer(s) (either 1, 2, 3, or 4) specifying the alignment of the character vector(s) given via `ilab` (2 means right, 4 mean left aligned). The default is to center the labels.

`transf` 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., `transf=exp`). 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 x axis labels and annotations (e.g., `transf=exp`). Defaults to `FALSE`, which means that no transformation is used.

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

`addrows` number of extra rows that should be added at the bottom of the forest plot (default is 0).

`efac` 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.

`psize` an optional vector with point sizes for the observed effect sizes or outcomes. Default is 1.

`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 x axis labels. If `NULL` (default), the function tries to set this to a sensible value.

`...` 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 (`alim`). 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@www.wvbauer.com⟩; http://www.wvbauer.com/

See Also

forest, cumul

Examples

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

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

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

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

### meta-analysis of the (log) risk ratios using the Mantel-Haenszel method
res <- rma.mh(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, measure="RR")
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, annotate=TRUE, 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, atransf=FALSE, targs=NULL,
       addrows=0, efac=1, pch=15, psize=NULL,
       cex=NULL, cex.lab=NULL, cex.axis=NULL, ...)
```

Arguments

`x` a vector of length k with the observed effect sizes or outcomes.

`vi` a vector of length k with the corresponding sampling variances.

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

`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 and corresponding labels 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 (default is 95).

`digits` integer value specifying the number of decimal places to which the tick mark labels and 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 (`NULL` by default, which suppresses the title).

`slab` 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`.

`ilab` an optional vector or matrix of character strings providing additional information about the studies.

`ilab.xpos` numerical value(s) specifying the x axis position(s) of the character vector(s) given via `ilab`.

`ilab.pos` integer(s) (either 1, 2, 3, or 4) specifying the alignment of the character vector(s) given via `ilab` (2 means right, 4 mean left aligned). The default is to center the labels.

`subset` 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.

`transf` an optional argument specifying the name of a function that should be used to transform the observed effect sizes and corresponding confidence interval bounds (e.g., `transf=exp`). 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 x axis labels and annotations (e.g., `transf=exp`). Defaults to `FALSE`, which means that no transformation is used.

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

`addrows` number of extra rows that should be added at the bottom of the forest plot (default is 0).

`efac` 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.

`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 inverse of the sampling variances.

`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 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.

With the `transf` argument, the observed effect sizes 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. 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 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 (`alim`). 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@www.wvbauer.com⟩; http://www.wvbauer.com/

See Also

forest, addpoly

Examples

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

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

### forest plot of the observed risk ratios
forest(dat$yi, dat$vi, slab=paste(dat.bcg$author, dat$year),
       atransf=exp, xlab="Risk Ratio (log scale)")
```

---

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, 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, atransf=FALSE, targs=NULL, addrows=0,
       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 or fitted values should be added to the plot. See 'Details'.
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 and corresponding labels 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 and 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 (`NULL` by default, which suppresses the 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`.

`ilab.pos` integer(s) (either 1, 2, 3, or 4) specifying the alignment of the character vector(s) given via `ilab` (2 means right, 4 mean left aligned). The default is to center the labels.

`order` an optional character string specifying how the studies should be ordered. See 'Details'.

`transf` 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., `transf=exp`). 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 x axis labels and annotations (e.g., `transf=exp`). Defaults to `FALSE`, which means that no transformation is used.

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

`addrows` number of extra rows that should be added at the bottom of the forest plot (default is 0).

`efac` 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.

`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 inverse of the sampling variances.

`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 x axis labels. If `NULL` (default), the function tries to set this to a sensible value.

`col` character string specifying the name of a color to use for the fitted values (`"darkgray"` by default).

`border` character string specifying the name of a color to use for the border of the fitted values (`"darkgray"` by default).

`...` other arguments.

Details

The plot shows the individual observed effect sizes or outcomes with corresponding confidence intervals. For fixed- and random-effects models, a polygon is added to the bottom of the forest plot, showing the summary estimate based on the model. For models involving moderators, the fitted values are added to the plot.

With the `transf` argument, the observed effect sizes, summary estimate, fitted values, and confidence interval bounds can be transformed with an arbitrary function. For example, if log odds ratios were analyzed, 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. 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 possible to set `order` equal to a vector with indices giving the desired order.

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 (`alim`). 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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

See Also

`forest`, `addpoly`

Examples

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

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

### several forest plots illustrating the use of various arguments
forest(res, xlab="Log Relative Risk")
forest(res, xlab="Log Relative Risk", order=order(dat.bcg$ablat))
forest(res, transf=exp, alim=c(0,6), steps=4, xlim=c(-8,12),
       xlab="Relative Risk", refline=1)
forest(res, atransf=exp, at=log(c(.05,.25,1,4,12)), xlim=c(-8,6),
       xlab="Relative Risk (log scale)", order="prec")
forest(res, slab=paste(dat.bcg$author, ", ", dat.bcg$year, sep=""),
       xlim=c(-14,6), ylim=c(-1.5,15.5), at=log(c(.05,.25,1,4)),
       atransf=exp, xlab="Relative Risk (log scale)",
       ilab=cbind(dat.bcg$tpos, dat.bcg$tneg, dat.bcg$cpos, dat.bcg$cneg),
       ilab.xpos=c(-8.5,-7,-5,-3.5), cex=.7)
text(c(-8.5,-7,-5,-3.5), 14.5, c("TB+", "TB-", "TB+", "TB-"), cex=.7)
text(c(-7.75,-4.25),     15.5, c("Vaccinated", "Control"),    cex=.7)
text(-14,                14.5, "Author(s) and Year",   pos=4, cex=.7)
text(6,                  14.5, "Observed RR [95% CI]", pos=2, cex=.7)

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

### forest plots with observed and fitted values
forest(res, xlim=c(-9,5), order="fit", xlab="Relative Risk (log scale)", cex=.8,
       ilab=dat.bcg$ablat, ilab.xpos=-4, atransf=exp, at=log(c(.05,.25,1,4)))
```

```
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)
```

fsn *Fail-Safe N*

Description

Function to calculate the fail-safe N.

Usage

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

Arguments

`yi`	a vector with the observed effect sizes or outcomes.
`vi`	a vector with the corresponding sampling variances.
`sei`	a vector with the corresponding standard errors. (note: only one of the two, `vi` or `sei`, needs to be specified)
`data`	an optional data frame containing the variables given to the arguments above.
`type`	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.
`alpha`	target alpha level to use for the Rosenthal and Rosenberg methods (.05 by default).
`target`	target average effect size to use for the Orwin method. If `NULL`, then the target average effect size will be equal to the observed average effect size divided by 2.
`digits`	an integer specifying the number of decimal places to which the printed results should be rounded (default is 4).
`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.

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 formated 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@www.wvbauer.com⟩; `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.

See Also

`ranktest`, `trimfill`

Examples

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

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

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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

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, level=x$level, digits=3, 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.

`ylab` title for the y axis.

`steps` the number of tick marks and corresponding labels for the y axis (default is 5).

`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’.

`digits` integer value specifying the number of decimal places to which the tick mark labels on the y axis should be rounded (default is 3).

`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 "rstandard" (default) or "rstudent" 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 level 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 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 $\pm 1.96 SE$, 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@www.wvbauer.com⟩; `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.

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 risk ratios using a random-effects model
```

```
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg,
           data=dat.bcg, measure="RR", method="REML")
funnel(res)

### 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, refline=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@www.wvbauer.com⟩; http://www.wvbauer.com/

See Also

rma.uni, influence.rma.uni

Examples

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

### meta-analysis of the log risk ratios 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 case:

- externally standardized residual,
- DFFITS value,
- Cook's distance,
- covariance ratio,
- the leave-one-out amount of (residual) heterogeneity,
- the leave-one-out test statistic 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 (the hat values and weights are the same for models without moderators, but will differ once moderators are included).

For details on externally standardized residuals, see `rstudent.rma.uni`.

For the meta-analytic models underlying the `rma.uni` function, the DFFITS value for the i^{th} case is here defined as the difference between the predicted value based on the fitted model using the complete dataset and the predicted value based on the fitted model using the dataset with the i^{th} case removed, divided by the standard error of the predicted value based on the fitted model using the dataset with the i^{th} case removed.

Cook's distance is here defined as $(\hat{\beta} - \hat{\beta}_{(i)})'\hat{\Sigma}^{-1}(\hat{\beta} - \hat{\beta}_{(i)})$, where $\hat{\beta}$ is the vector of parameter estimates based on the complete dataset, $\hat{\beta}_{(i)}$ is the vector of parameter estimates based on the dataset with the i^{th} case removed, and $\hat{\Sigma}$ is the variance-covariance matrix of the parameter estimates based on the full dataset.

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} case removed divided by the determinant of the variance-covariance matrix of the parameter estimates based on the complete dataset.

The leave-one-out amount of (residual) heterogeneity is the estimated value of τ^2 based on the dataset with the i^{th} case removed. 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} case removed.

Finally, the DFBETAS values for the i^{th} case are defined as the difference between the parameter estimates based on the complete dataset and the parameter estimates based on the dataset with the i^{th} case removed, divided by the standard error of the parameter estimates based on the dataset with the i^{th} case removed.

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} case from the dataset. This will result in `NA` values for that case.

Certain relationships between the leave-one-out diagnostics and the (internally or externally) standardized residuals (Belsley, Kuh, & Welsch, 1980; Cook & Weisberg, 1982) no longer hold for the meta-analytic models. Maybe there are other relationships. These remain to be determined.

Author(s)

Wolfgang Viechtbauer; ⟨wvb@www.wvbauer.com⟩; `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.

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 risk ratios 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)
```

`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@www.wvbauer.com⟩; http://www.wvbauer.com/

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

`x`	an object of class `"rma.mh"` or `"rma.peto"`.
`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 risk ratios (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 "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 formated and printed with print.list.rma.

Author(s)

Wolfgang Viechtbauer; ⟨wvb@www.wvbauer.com⟩; http://www.wvbauer.com/

See Also

leave1out

Examples

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

### meta-analysis of the (log) risk ratios 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=NULL, ...)
```

Arguments

`x`	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).
`transf`	an optional argument specifying the name of a function that should be used to transform the model coefficients 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

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:

`estimate`	estimated coefficients of the model.
`se`	standard errors of the coefficients. `NA` if `transf` is used to transform the coefficients.
`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.
`tau2`	estimated amounts of (residual) heterogeneity (only for random-effects models).
`I2`	values of I^2 (only for random-effects models).
`H2`	values of H^2 (only for random-effects models).

The `"list.rma"` object is formated 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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

See Also

`leave1out`, `influence.rma.uni`

Examples

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

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

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

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

`logLik.rma` *Log-Likelihood of rma Objects*

Description

The function extracts the (restricted) log likelihood for objects of class `"rma"`.

Usage

```
## S3 method for class 'rma':
logLik(object, REML=NULL, ...)
```

Arguments

`object`	an object of class `"rma"`.
`REML`	logical indicating whether the regular or restricted log likelihood should be returned. When `NULL` (default), the regular log likelihood is returned, unless restricted maximum likelihood estimation was used to fit the model.
`...`	other arguments.

Value

The (restricted) log likelihood of the model evaluated at the estimated coefficient(s).

Note

For a more complete set of fit statistics and information criteria, see `fitstats.rma`.

Author(s)

Wolfgang Viechtbauer; ⟨wvb@www.wvbauer.com⟩; http://www.wvbauer.com/

See Also

`fitstats.rma`, `rma.uni`, `rma.mh`, `rma.peto`

Examples

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

### meta-analysis of the log risk ratios 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")
logLik(res)
```

`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, 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`).

`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 $1/2$.
- 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, except for their scaling, the hat values and weights will be equal to each other in models without moderators.

The chosen cut-offs are (somewhat) arbitrary. Substantively informed judgment should always be used when examining the influence of each case on the results.

Author(s)

Wolfgang Viechtbauer; ⟨wvb@www.wvbauer.com⟩; `http://www.wvbauer.com/`

See Also

`influence.rma.uni`

Examples

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

### meta-analysis of the log risk ratios 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@www.wvbauer.com⟩; http://www.wvbauer.com/

See Also

forest, funnel, radial, qqnorm.rma.uni

Examples

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

### meta-analysis of the log risk ratios 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.uni *Predicted Values for rma.uni Objects*

Description

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

Usage

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

Arguments

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

`newmods` an optional vector or matrix specifying the values of the moderator values for which the predicted values should be calculated. See ‘Details’.

`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 l new studies can be specified by setting `newmods` equal to an $l \times q$ matrix with the corresponding new moderator values. An example is 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 `"list.rma"`. The object is a list containing the following components:

`pred` predicted value(s).

`se` corresponding standard error(s).

`ci.lb` lower bound of the confidence interval(s).

`ci.ub` upper bound of the confidence interval(s).

`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).

`...` some additional elements/values.

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

Note

The predicted values are based only on the fixed effects in 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`.

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@www.wvbauer.com⟩; http://www.wvbauer.com/

References

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

Raudenbush, S. W. (1994) Random effects models. In H. C. Cooper & L. V. Hedges (Eds.), *The handbook of research synthesis* (pp. 301–321). New York: Russell Sage Foundation.

See Also

fitted.rma, blup.rma.uni

Examples

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

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

### average risk ratio 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=ablat,
           measure="RR", data=dat.bcg, method="REML")

### predicted average risk ratios for given absolute latitude values
predict(res, transf=exp)

### predicted average risk ratios 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 risk ratios for 10 and 60 degrees latitude in 1950 and 1980
predict(res, newmods=cbind(c(10,60,10,60),c(1950,1950,1980,1980)), transf=exp)
```

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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

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@www.wvbauer.com⟩; http://www.wvbauer.com/

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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

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@www.wvbauer.com⟩; http://www.wvbauer.com/

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 unreleated 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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

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

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

`type` either `"rstandard"` (default) or `"rstudent"` indicating whether the usual or deleted residuals should be used in creating the plot. See 'Details'.

`pch` plotting symbol to use for the observed effect sizes or outcomes. By default, a solid circle is used. See `points` for other options.

`envelope` 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’.

`level` numerical value between 0 and 100 specifying the level of the pseudo confidence envelope (the default is to take the value from the object).

`bonferroni` logical indicating whether the bounds of the envelope should be Bonferroni corrected.

`reps` numerical value indicating the number of iterations to use for simulating the pseudo confidence envelope (default is 1000).

`smooth` logical indicating whether the results from the simulation should be smoothed (default is `TRUE`).

`bass` numerical value that controls the degree of smoothing (default is 0).

`...` 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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

References

Cook, R. D. & Weisberg, S. (1982) *Residuals and influence in regression*. London: Chapman and Hall.

See Also

`rma.uni`, `rma.mh`, `rma.peto`

Examples

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

### meta-analysis of the log risk ratios 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=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@www.wvbauer.com⟩; `http://www.wvbauer.com/`

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=NULL, 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 and corresponding labels 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	an optional integer value specifying the number of decimal places to which the y axis tick mark labels 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@www.wvbauer.com⟩; `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.

See Also

`rma.uni`, `rma.mh`, `rma.peto`,

Examples

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

### meta-analysis of the log risk ratios 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 of the log risk ratio from the 4th study
abline(a=0, b=c(-1.44155119))

### meta-analysis of the log risk ratios 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@www.wvbauer.com⟩; 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.

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 formated 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@www.wvbauer.com⟩; `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.

See Also

ranktest, regtest

Examples

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

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

### 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

x	an object of class "rma".
...	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@www.wvbauer.com⟩; http://www.wvbauer.com/

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=NULL, ...)
```

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 for the regression test. See 'Details'.
ni	the total sample size. Only needed when carrying out a regression test with either predictor="ni" or predictor="ninv" and the object x does not already contain this information.
...	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 size 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:

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

The results are formated 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@www.wvbauer.com⟩; `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.

See Also

`regtest`, `ranktest`

Examples

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

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

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

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

object	an object of class "rma" (for residuals).
model	an object of class "rma.uni", "rma.mh", or "rma.peto" (for rstandard and rstudent).
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 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:

`resid`	observed residuals (for `rstandard`) or deleted residuals (for `rstudent`).
`se`	corresponding standard errors.
`z`	standardized residuals (internally standardized for `rstandard` or externally standardized for `rstudent`).

The `"list.rma"` object is formated 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@www.wvbauer.com⟩; http://www.wvbauer.com/

References

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

See Also

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

Examples

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

### meta-analysis of the log risk ratios 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=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 for data from 2x2 tables via the Mantel-Haenszel method.

Usage

```
rma.mh(ai, bi, ci, di, n1i, n2i, data=NULL,
       slab=NULL, subset=NULL, measure="OR",
       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 variables given to the arguments above.

`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.

`measure` a character string indicating the outcome measure to use for the meta-analysis: the odds ratio (`"OR"`), relative risk (`"RR"`), or risk difference (`"RD"`).

`add` vector with two numbers indicating the amount to add to the cell frequencies of the 2x2 table. 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 `escalc` function for more details.

`to`	a character vector with two strings indicating when the values under `add` should be added. When set to `"all"`, the value of `add` is added to each cell of the 2x2 tables in all k tables. When set to `"only0"`, the value of `add` is added to each cell of the 2x2 tables with at least one cell equal to 0. When set to `"if0all"`, the value of `add` is added to each cell of the 2x2 tables in all k studies, but only when there is at least one 2x2 table with a zero entry. Setting `to="none"` or `add=0` has the same effect: No adjustment to the observed table frequencies is made. 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 `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 providing data in terms of 2x2 tables are assumed to be of the form:

	outcome 1	outcome 2	total
group 1	`ai`	`bi`	`n1i`
group 2	`ci`	`di`	`n2i`

where `ai`, `bi`, `ci`, and `di` denote the cell frequencies and `n1i` and `n2i` the row totals. For example, in a set of RCTs, group 1 and group 2 may refer to the treatment and placebo group, with outcome 1 denoting some event of interest 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 Mantel and Haenszel (1959), which can be used in combination with the odds ratio, risk ratio, or risk difference (`measure="OR"`, `"RR"`, or `"RD"`). 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 and relative risks, 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 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 also 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 outcome values 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 outcome values and when applying the Mantel-Haenszel method.

Value

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

`b` aggregated log odds ratio, log risk ratio, or risk difference.

`se` standard error of the aggregated value.

`zval` test statistics of the aggregated value.

`pval` p-value for the test statistic.

`ci.lb` lower bound of the confidence interval.

`ci.ub` upper bound of the confidence interval.

`QE` test statistic for the test of heterogeneity.

`QEp` 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 formated 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, and radial plots of the individual outcomes can be obtained with `forest.rma`, `funnel.rma`, and `radial.rma`. The `qqnorm.rma.mh` function provides a normal QQ plot 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 obervation at a time) can be obtained with `cumul.rma.mh`.

Other assessor functions include `coef.rma`, `vcov.rma`, and `logLik.rma`.

Author(s)

Wolfgang Viechtbauer; ⟨wvb@www.wvbauer.com⟩; `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.

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) risk ratios 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=NULL,
         slab=NULL, subset=NULL,
         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 variables given to the arguments above.

`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 numbers indicating the amount to add to the cell frequencies of the 2x2 table. 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. When set to `"all"`, the value of `add` is added to each cell of the 2x2 tables in all k tables. When set to `"only0"`, the value of `add` is added to each cell of the 2x2 tables with at least one cell equal to 0. When set to `"if0all"`, the value of `add` is added to each cell of the 2x2 tables in all k studies, but only when there is at least one 2x2 table with a zero entry. Setting `to="none"` or `add=0` has the same effect: No adjustment to the observed table frequencies is made. 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 providing data in terms of 2x2 tables are assumed to be of the form:

	outcome 1	outcome 2	total
group 1	`ai`	`bi`	`n1i`
group 2	`ci`	`di`	`n2i`

where `ai`, `bi`, `ci`, and `di` denote the cell frequencies and `n1i` and `n2i` the row totals. For example, in a set of RCTs, group 1 and group 2 may refer to the treatment and placebo group, with outcome 1 denoting some event of interest 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 also 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.

Value

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

`b`	aggregated log odds ratio.

`se` standard error of the aggregated value.

`zval` test statistics of the aggregated value.

`pval` p-value for the test statistic.

`ci.lb` lower bound of the confidence interval.

`ci.ub` upper bound of the confidence interval.

`QE` 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 unrestricted and restricted likelihood.

`...` some additional elements/values.

The results of the fitted model are neatly formated 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, and radial plots of the individual outcomes can be obtained with `forest.rma`, `funnel.rma`, and `radial.rma`. The `qqnorm.rma.peto` function provides a normal QQ plot 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 obervation at a time) can be obtained with `cumul.rma.peto`.

Other assessor functions include `coef.rma`, `vcov.rma`, and `logLik.rma`.

Author(s)

Wolfgang Viechtbauer; ⟨wvb@www.wvbauer.com⟩; `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.

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 General Linear (Mixed-Effects) Model*

Description

Function to fit the meta-analytic fixed- and random-effects models with or without moderators via the general 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, m1i, m2i, sd1i, sd2i,
        xi, mi, ri, ni, mods=NULL, data=NULL, intercept=TRUE, slab=NULL,
        subset=NULL, measure="GEN", add=1/2, to="only0", vtype="LS",
        method="REML", weighted=TRUE, level=95, digits=4, btt=NULL,
        tau2=NULL, knha=FALSE, control=list())
rma(yi, vi, sei, ai, bi, ci, di, n1i, n2i, m1i, m2i, sd1i, sd2i,
        xi, mi, ri, ni, mods=NULL, data=NULL, intercept=TRUE, slab=NULL,
        subset=NULL, measure="GEN", add=1/2, to="only0", vtype="LS",
        method="REML", weighted=TRUE, level=95, digits=4, btt=NULL,
        tau2=NULL, knha=FALSE, control=list())
```

Arguments

`yi` vector of length k with the observed effects sizes or outcomes. See 'Details'.

`vi` vector of length k with the corresponding sampling variances. See 'Details'.

`sei` vector of length k with the corresponding standard errors. See 'Details'.

`ai` see below and the documentation of the `escalc` function for more details.

`bi` see below and the documentation of the `escalc` function for more details.

`ci` see below and the documentation of the `escalc` function for more details.

`di` see below and the documentation of the `escalc` function for more details.

`n1i` see below and the documentation of the `escalc` function for more details.

`n2i` see below and the documentation of the `escalc` function for more details.

`m1i` see below and the documentation of the `escalc` function for more details.

`m2i` see below and the documentation of the `escalc` function for more details.

`sd1i` see below and the documentation of the `escalc` function for more details.

`sd2i` see below and the documentation of the `escalc` function for more details.

`xi` see below and the documentation of the `escalc` function for more details.

`mi` see below and the documentation of the `escalc` function for more details.

`ri` see below and the documentation of the `escalc` function for more details.

`ni` 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.

`data` an optional data frame containing the variables given to the arguments above.

`intercept` logical, indicating whether an intercept term should be added to the model (default is `TRUE`).

`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.

`measure` a character string indicating the type of data supplied to the function. When `measure="GEN"` (default), the observed effect sizes or outcomes and corresponding sampling variances (or standard errors) should be supplied to the function via the `yi`, `vi`, and `sei` arguments (only one of the two, `vi` or `sei`, needs to be specified). Alternatively, one can set `measure` to one of the effect size or outcome measures described under the documentation for the `escalc` function and supply the needed data via the appropriate arguments.

`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 `method="FE"`. Random/mixed-effects models are fitted by setting `method` 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.

`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).

`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’.

`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 risk ratios, risk differences, mean differences, standardized mean differences, raw correlation coefficients, correlation coefficients transformed with Fisher's r-to-z transformation, and so on). Simply supply the observed outcomes via the `yi` argument and the corresponding sampling variances via the `vi` argument (or supply the standard errors, the square root of sampling variances, via the `sei` argument). In this case, one should set `measure="GEN"` (the default).

Alternatively, the function can automatically calculate many effect size or outcome measures (and the corresponding sampling variances) when supplied with the needed data. The `escalc` function describes which measures are currently implemented and what data/arguments should then be specified. The `measure` argument should then be set to the desired 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 a matrix with k rows and p' columns (e.g., using `mods = cbind(mod1, mod2, mod3)`, where `mod1`, `mod2`, `mod3` correspond to the names of the variables for the three moderator variables). 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. The intercept is automatically included in the model, unless `intercept=FALSE` is used.

Omnibus Test of Parameters

In models with more than one independent variable, an omnibus test of all the regression coefficients is conducted that excludes the intercept (the first coefficient) if the option `intercept=TRUE` is used (which is the default). If `intercept=FALSE`, then the omnibus test includes all of the coefficients in the model. Alternatively, one can 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.

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 have to do the dummy coding yourself or use the `model.matrix` function to do the coding for you. An example is shown 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 (p being the total number of coefficients in the model), while the omnibus test statistic then uses an F-distribution with m and $k - p$ degrees of freedom. The Knapp and Hartung (2003) method 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:

`b` estimated coefficients of the model.

`se` standard errors of the coefficients.

`zval` test statistics of the coefficients.

`pval` p-values for the test statistics.

`ci.lb` lower bound of the confidence intervals for the coefficients.

`ci.ub` upper bound of the confidence intervals for the coefficients.

`vb` variance-covariance matrix of the estimated coefficients.

`tau2` estimated amount of (residual) heterogeneity. Always `0` when `method="FE"`.

`se.tau2` 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 formated 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`.

Predicted/fitted values can be obtained with `predict.rma.uni` 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 `cint.rma.uni`.

Forest, funnel, and radial plots (the latter only for models without moderators) are drawn with `forest.rma`, `funnel.rma`, and `radial.rma`. The `qqnorm.rma.uni` function provides a normal QQ plot 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, a cumulative meta-analysis (i.e., adding one obervation at a time) can be obtained with `cumul.rma.uni`.

Other assessor functions include `coef.rma`, `vcov.rma`, `logLik.rma`, and `hatvalues.rma.uni`.

Note

The ML, REML, and EB estimators of τ^2 are obtained via 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 50 by default. 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)`. 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 `vi` 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 arc-sine 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) method 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@www.wvbauer.com⟩; 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. (1994) Random effects models. In H. C. Cooper & L. V. Hedges (Eds.), *The handbook of research synthesis* (pp. 301–321). 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.

Examples

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

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

### 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")

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

### 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))

### use model.matrix() to code the factor and set up the design matrix
### careful: X already includes the intercept, so need to use intercept=FALSE
```

```
X <- model.matrix(~ factor(alloc) + year + ablat, data=dat)
rma(yi, vi, mods=X, intercept=FALSE, data=dat, method="REML", btt=c(2,3))
```

`transf` *Transformation Functions*

Description

A set of transformation functions useful for meta-analyses.

Usage

```
transf.rtoz(x, ...)
transf.ztor(x, ...)
transf.logit(x, ...)
transf.ilogit(x, ...)
transf.ztor.int(x, targs, ...)
transf.exp.int(x, targs, ...)
```

Arguments

`x` a vector of values to be transformed.

`targs` a list with additional arguments for the transformation function. See 'Details'.

`...` other arguments.

Details

The following transformation functions are currently implemented:

- `transf.rtoz`: Fisher's r-to-z transformation.
- `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.ztor.int`: integral transformation method for the z-to-r transformation.
- `transf.exp.int`: integral transformation method for the exponential transformation.

The integral transformation method for a transformation function $h(z)$ integrates $h(z) \times f(z)$ over the limits `targs$lower` and `targs$upper`, where $f(z)$ is the density of a normal distribution with mean equal to `x` and variance equal to `targs$tau2`.

Value

A vector with the transformed values.

Author(s)

Wolfgang Viechtbauer; ⟨wvb@www.wvbauer.com⟩; http://www.wvbauer.com/

Examples

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

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

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

### average risk ratio with 95% CI using 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@www.wvbauer.com⟩; `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.

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, estimator="L0", side=NULL, maxit=50, verbose=FALSE, ...)
```

Arguments

`x`	an object of class `"rma.uni"`.
`estimator`	either `"L0"` or `"R0"` indicating the estimator to use for estimating the number of missing studies.
`side`	either `"left"`, `"right"`, or `NULL` indicating on which side of the funnel plot the missing studies should be imputed. If `NULL`, the side is chosen within the function depending on the results of Egger's regression test (see `regtest` for details on this test).
`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 object also contains a vector of dummy codes (called `fill`), indicating which of the data are the observed (0) and the augmented (1) data. Calling `funnel.rma` on the object provides a funnel plot of the observed and augmented data.

Author(s)

Wolfgang Viechtbauer; ⟨wvb@www.wvbauer.com⟩; `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.

See Also

`funnel.rma`

Examples

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

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

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

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@www.wvbauer.com⟩; http://www.wvbauer.com/

See Also

rma.uni, rma.mh, rma.peto

Examples

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

### meta-analysis of the log risk ratios 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)
```

Index