Package 'metafor'

February 15, 2013

Version 1.7-0

Date 2013-02-05

Title Meta-Analysis Package for R

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

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

Depends R (>= 2.15.2), stats, Formula

Suggests lme4, polycor, BiasedUrn, survival, Epi

Description The metafor package provides a comprehensive collection of functions for conducting meta-analyses in R. The package includes functions to calculate various effect size or outcome measures, fit fixed-, random-, and mixed-effects models to such data, carry out moderator and meta-regression analyses, and create various types of meta-analytical plots (e.g., forest,funnel, radial, and L'Abbe plots). For meta-analyses of 2x2 tables, proportions, incidence rates, and incidence rate ratios, the package also provides functions that implement specialized methods, including the Mantel-Haenszel method,Peto's method, and a variety of suitable generalized linear (mixed-effects) models (i.e., mixed-effects (conditional) logistic and Poisson regression models).

License GPL (>= 2)

ByteCompile TRUE

BuildVignettes FALSE

URL http://www.metafor-project.org/, http://www.wvbauer.com

Repository CRAN

Date/Publication 2013-02-06 11:56:38

NeedsCompilation no

R topics documented:

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

Description

The **metafor** package provides a comprehensive collection of functions for conducting meta-analyses in R. The package includes functions for calculating various effect size or outcome measures frequently used in meta-analyses (e.g., risk differences, risk ratios, odds ratios, standardized mean differences, Fisher's r-to-z-transformed correlation coefficients) and then allows the user to fit fixed-, random-, and mixed-effects models to these data. By including study-level covariates ('moderators') in these models, so-called 'meta-regression' analyses can be carried out. For meta-analyses of 2x2 tables, proportions, incidence rates, and incidence rate ratios, the package also provides functions that implement specialized methods, including the Mantel-Haenszel method, Peto's method, and a variety of suitable generalized linear (mixed-effects) models (i.e., mixed-effects (conditional) logistic and Poisson regression models).

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

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

The rma.uni Function

The various meta-analytic models that are typically used in practice are special cases of the general linear (mixed-effects) model. The `rma.uni` function (with alias `rma`) provides a general framework for fitting such models. The function can be used in conjunction with any of the usual effect size or outcome measures used in meta-analyses (e.g., log odds ratios, log relative risks, risk differences, mean differences, standardized mean differences, raw correlation coefficients, correlation coefficients transformed with Fisher's r-to-z transformation, and so on). For details on these effect size or outcome measures, see the documentation of the `escalc` function. The notation and models underlying the `rma.uni` function are explained below.

For a set of $i = 1, \ldots, k$ 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, such that

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

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

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

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

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

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

Moderators can be included in the fixed-effects model, yielding a **fixed-effects with moderators model**. Again, since the model conditions on the set of k studies included in the meta-analysis,

the regression coefficients from the fitted model estimate the weighted least squares relationship between the true effects and the moderator variables within the set of k studies included in the meta-analysis (again using weights equal to $w_i = 1/v_i$). The (unweighted) least squares relationship between the true effects and the moderator variables can be estimated when using the unweighted estimation method.

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

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

where $u_i \sim N(0, \tau^2)$ and $e_i \sim N(0, v_i)$. The fitted model provides an estimate of μ and τ^2. Consequently, the random-effects model provides an *unconditional inference* about the average 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. Such a meta-regression model can also be written as

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

where $u_i \sim N(0, \tau^2)$ and x_{ij} denotes the value of the j^{th} moderator variable for the i^{th} study. Therefore, β_j denotes how the average true effect or outcome changes for a one unit increase in x_{ij} and the model intercept β_0 denotes the average true effect or outcome when the values of all moderator variables are equal to zero (note that there are $p = p' + 1$ coefficients in the model when the model intercept is included). The coefficients from the fitted model therefore 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 that is not accounted for by the moderators included in the model).

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

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 in all k studies). In other words, fixed-effects models provide perfectly valid inferences under heterogeneity, as long as one is restricting these inferences to the set of studies

included in the meta-analysis and one realizes that the model does not provide an estimate of θ, but of θ_w or θ_u. On the other hand, the random-effects model provides an inference about the average effect in the entire population of studies from which the included studies are assumed to be a random selection.

In the special case that the true effects are actually homogeneous, the distinction between fixed- and random-effects models disappears, since homogeneity implies that $\mu = \bar{\theta}_w = \bar{\theta}_u \equiv \theta$. However, since there is no infallible method to test whether the true effects are really homogeneous or not, a researcher should decide on the type of inference desired before examining the data and choose the model accordingly. For more details on the distinction between fixed- and random-effects models, see Laird and Mosteller (1990) and Hedges and Vevea (1998).

The rma.mh Function

The Mantel-Haenszel method provides an alternative approach for fitting the fixed-effects model when dealing with studies providing data in the form of 2x2 tables or in the form of event counts for two groups (Mantel & Haenszel, 1959). The method is particularly advantageous when aggregating a large number of studies with small sample sizes (the so-called sparse data or increasing strata case). The Mantel-Haenszel method is implemented in the `rma.mh` function. It can be used in combination with odds ratios, relative risks, risk differences, and incidence rate ratios. The Mantel-Haenszel method is always based on a weighted estimation approach.

The rma.peto Function

Yet another method that can be used in the context of a meta-analysis of 2x2 table data 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.

The rma.glmm Function

Dichotomous and event count data (based on which one can calculate effect size or outcome measures, such as odds ratios, incidence rate ratios, proportions, and incidence rates) are often assumed to arise from binomial and Poisson distributed data. Meta-analytic models that are directly based on such distibutions have been developed and are implemented in the `rma.glmm` function. The collection of models implemented in that function are essentially special cases of generalized linear (mixed-effects) models (i.e., mixed-effects logistic and Poisson regression models). For 2x2 table data, a mixed-effects conditional logistic model (based on the non-central hypergeometric distribution) is also implemented. Random/mixed-effects models with dichotomous data are often referred to as 'binomial-normal' models in the meta-analytic literature. Analogously, for event count data, such models could be referred to as 'Poisson-normal' models.

Future Plans and Updates

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

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

Citing the Package

To cite the package, please use the following reference:

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

Getting Started with the Package

The paper mentioned above is a good starting place for those interesting in using the **metafor** package. The purpose of the article is to provide a general overview of the package and its capabilities (as of version 1.4-0). Not all of the functions and options are described in the paper, but it should provide a pretty thorough introduction. The paper can be freely downloaded from the URL given above.

In addition to reading the paper, carefully read this page and then the help pages for the `escalc` and the `rma.uni` functions (or the `rma.mh`, `rma.peto`, and `rma.glmm` functions if you intend to use these methods). The help pages for these functions provide links to many additional functions, which can be used after fitting a model.

Finally, additional information about the package and more detailed examples can be found on the package homepage at http://www.metafor-project.org/.

Author(s)

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

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

References

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

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

Hedges, L. V., & Vevea, J. L. (1998). Fixed- and random-effects models in meta-analysis. *Psychological Methods*, **3**, 486–504.

Laird, N. M., & Mosteller, F. (1990). Some statistical methods for combining experimental results. *International Journal of Technology Assessment in Health Care*, **6**, 5–30.

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

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

Yusuf, S., Peto, R., Lewis, J., Collins, R., & Sleight, P. (1985). Beta blockade during and after myocardial infarction: An overview of the randomized trials. *Progress in Cardiovascular Disease*, **27**, 335–371.

`addpoly` *Add Polygons to Forest Plot*

Description

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

Usage

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

Arguments

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

Details

Currently, methods exist for two types of situations.

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

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

Author(s)

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

References

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

See Also

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

addpoly.default *Add Polygons to Forest Plot*

Description

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

Usage

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

Arguments

x	vector with the values at which the polygons should be drawn.
vi	vector with the corresponding variances.
sei	vector with the corresponding standard errors. (note: only one of the two, vi or sei, needs to be specified)
ci.lb	vector with the corresponding lower confidence intervals bounds. Not needed if vi or sei is specified. See 'Details'.
ci.ub	vector with the corresponding upper confidence intervals bounds. Not needed if vi or sei is specified. See 'Details'.
rows	vector specifying the rows (or more generally, the horizontal positions) for plotting the polygons (defaults is -1). Can also be a single value specifying the row (horizontal position) of the first polygon (the remaining polygons are then plotted below this starting row).
level	numerical value between 0 and 100 specifying the confidence interval level (default is 95).
digits	integer 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.
transf	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	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	optional symbol expansion factor. If unspecified, the function tries to set this to a sensible value.
...	other arguments.

Details

The function can be used to add one or more polygons to an existing forest plot created with the forest function. For example, summary estimates based on a subgrouping of the studies or from models involving moderators can be added to the plot this way. See examples below.

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

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

Author(s)

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

References

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

See Also

forest.rma, forest.default

Examples

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

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

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

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

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

### forest plot with subgrouping of studies and summaries per subgroup
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, measure="RR",
           slab=paste(author, year, sep=", "), method="REML")
forest(res, xlim=c(-16, 6), at=log(c(.05, .25, 1, 4)), atransf=exp,
       ilab=cbind(dat.bcg$tpos, dat.bcg$tneg, dat.bcg$cpos, dat.bcg$cneg),
       ilab.xpos=c(-9.5,-8,-6,-4.5), cex=.75, ylim=c(-1, 27),
       order=order(dat.bcg$alloc), rows=c(3:4,9:15,20:23),
       mlab="RE Model for All Studies")
op <- par(cex=.75, font=4)
text(-16, c(24,16,5), c("Systematic Allocation", "Random Allocation",
                        "Alternate Allocation"), pos=4)
par(font=2)
text(c(-9.5,-8,-6,-4.5), 26, c("TB+", "TB-", "TB+", "TB-"))
text(c(-8.75,-5.25),     27, c("Vaccinated", "Control"))
text(-16,                 26, "Author(s) and Year",     pos=4)
text(6,                   26, "Relative Risk [95% CI]", pos=2)
par(op)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, measure="RR",
           subset=(alloc=="systematic"), method="REML")
addpoly(res, row=18.5, cex=.75, atransf=exp, mlab="RE Model for Subgroup")
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, measure="RR",
           subset=(alloc=="random"), method="REML")
addpoly(res, row=7.5, cex=.75, atransf=exp, mlab="RE Model for Subgroup")
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, measure="RR",
           subset=(alloc=="alternate"), method="REML")
addpoly(res, row=1.5, cex=.75, atransf=exp, mlab="RE Model for Subgroup")
```

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

Description

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

Arguments

x	an object of class "rma".
row	value specifying the row (or more generally, the horizontal position) for plotting the polygon (default is -2).
level	numerical value between 0 and 100 specifying the confidence interval level (the default is to take the value from the object).
digits	integer 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. If unspecified, the function sets a default label.
transf	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	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	optional symbol expansion factor. If unspecified, the function tries to set this to a sensible value.
...	other arguments.

Details

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

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

Author(s)

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

References

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

See Also

forest.rma, forest.default

Examples

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

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

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

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

### add summary estimate from the random-effects model to forest plot
addpoly(res, 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	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. This can be regarded as a pseudo-R^2 statistic (Raudenbush, 2009).

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

References

Hardy, R. J., & Thompson, S. G. (1996). A likelihood approach to meta-analysis with random effects. *Statistics in Medicine*, **15**, 619–629.

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

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

Viechtbauer, W. (2007). Hypothesis tests for population heterogeneity in meta-analysis. *British Journal of Mathematical and Statistical Psychology*, **60**, 29–60.

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

See Also

rma.uni, print.anova.rma.uni

Examples

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

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

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

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

anova(res1, res2)
```

---

blup *Best Linear Unbiased Predictions*

Description

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

Usage

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

Arguments

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

Details

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

Value

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

Author(s)

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

References

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

See Also

blup.rma.uni

---

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, ...)
```

Arguments

x	an object of class "rma.uni".
level	numerical value between 0 and 100 specifying the prediction interval level (the default is to take the value from the object).
digits	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	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.

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

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

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

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

Author(s)

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

References

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

Raudenbush, S. W., & Bryk, A. S. (1985). Empirical Bayes meta-analysis. *Journal of Educational Statistics*, **10**, 75–98.

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

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

See Also

rma.uni, predict.rma, fitted.rma

Examples

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

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

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

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

---

coef.permutest.rma.uni

Model Coefficients for permutest.rma.uni Objects

Description

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

Usage

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

Arguments

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

Value

A data frame with the following elements:

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

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

Author(s)

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

References

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

See Also

permutest.rma.uni, rma.uni

Examples

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

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

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

coef.rma *Model Coefficients for rma Objects*

Description

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

Usage

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

Arguments

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

Value

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

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

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

Author(s)

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

References

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

See Also

rma.uni, rma.mh, rma.peto, rma.glmm

Examples

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

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

---

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

Arguments

object	an object of class "rma.uni", "rma.mh", or "rma.peto". The method is not yet implemented for objects of class "rma.glmm".
parm	this argument is here for compatability with the generic function confint, but is ignored.
level	numerical value between 0 and 100 specifying the confidence interval level (the default is to take the value from the object).
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).
digits	integer specifying the number of decimal places to which the results should be rounded (the default is to take the value from the object).
verbose	logical indicating whether output should be generated for the iterative algorithm used as part of the Q-profile method (default is FALSE). See 'Note'.
control	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.

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

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

estimate	estimate of the model coefficient 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.

The results are formated and printed with the print.confint.rma function.

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. If verbose=TRUE, output is generated for the iterative algorithm.

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

Usually, the estimate of τ^2 from the random/mixed-effects model will fall within the confidence interval provided by the Q-profile method. However, this is not guaranteed. Depending on the method used to estimate τ^2 and the width of the confidence interval, it can happen in rare cases that the confidence interval does not actually contain the estimate (trying to explain this to reviewers can be tricky). However, using the empirical Bayes estimator of τ^2 when fitting the model (i.e., using `method="EB"`) guarantees that the estimate of τ^2 falls within the confidence interval.

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

Author(s)

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

References

Hartung, J., & Knapp, G. (2005). On confidence intervals for the among-group variance in the one-way random effects model with unequal error variances. *Journal of Statistical Planning and Inference*, **127**, 157–177.

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

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

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

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

Yusuf, S., Peto, R., Lewis, J., Collins, R., & Sleight, P. (1985). Beta blockade during and after myocardial infarction: An overview of the randomized trials. *Progress in Cardiovascular Disease*, **27**, 335–371.

See Also

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

Examples

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

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

### confidence interval for the total amount of heterogeneity
```

```
confint(res)

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

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

---

cumul *Cumulative Meta-Analysis*

Description

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

Usage

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

Arguments

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

Details

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

Author(s)

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

References

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

Lau, J., Schmid, C. H., & Chalmers, T. C. (1995). Cumulative meta-analysis of clinical trials builds evidence for exemplary medical care. *Journal of Clinical Epidemiology*, **48**, 45–57.

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

See Also

cumul.rma.uni, cumul.rma.mh, cumul.rma.peto

---

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

Description

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

Usage

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

Arguments

x	an object of class "rma.mh" or "rma.peto".
order	optional vector with indices giving the desired order for the cumulative meta-analysis.
digits	integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
transf	logical indicating whether odds ratios or relative risks (and the corresponding confidence interval bounds) should be given in the transformed (meaning: raw) units or in terms of log units (the default).
...	other arguments.

Value

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

estimate	estimated coefficients of the model.
se	standard errors of the coefficients. NA if 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.
QE	test statistics for the tests of heterogeneity.
QEp	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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

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

Lau, J., Schmid, C. H., & Chalmers, T. C. (1995). Cumulative meta-analysis of clinical trials builds evidence for exemplary medical care. *Journal of Clinical Epidemiology*, **48**, 45–57.

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

See Also

cumul, forest.cumul.rma

Examples

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

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

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

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

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

---

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

Description

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

Usage

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

Arguments

x	an object of class "rma.uni".
order	optional vector with indices giving the desired order for the cumulative meta-analysis.
digits	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	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.
QE	test statistics for the tests of heterogeneity.
QEp	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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

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

Lau, J., Schmid, C. H., & Chalmers, T. C. (1995). Cumulative meta-analysis of clinical trials builds evidence for exemplary medical care. *Journal of Clinical Epidemiology*, **48**, 45–57.

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

See Also

cumul, forest.cumul.rma

Examples

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

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

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

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

---

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

Description

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

Usage

```
dat.bonett2010
```

Format

The data frame contains the following columns:

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

Details

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

Source

Bonett, D. G. (2010). Varying coefficient meta-analytic methods for alpha reliability. *Psychological Methods*, **15**, 368–385.

References

Hakstian, A. R., & Whalen, T. E. (1976). A k-sample significance test for independent alpha coefficients. *Psychometrika*, **41**, 219–231.

Examples

```
### load data
data(dat.bonett2010)

### meta-analysis using the raw alpha values
res <- rma(measure="ARAW", ai=ai, mi=mi, ni=ni, data=dat.bonett2010)
res

### meta-analysis using transformed alpha values (using the
### transformation suggested by Hakstian & Whalen, 1976)
res <- rma(measure="AHW", ai=ai, mi=mi, ni=ni, data=dat.bonett2010)
res
predict(res, transf=transf.iahw)

### meta-analysis using transformed alpha values (using the
### transformation suggested by Bonett, 2002)
res <- rma(measure="ABT", ai=ai, mi=mi, ni=ni, data=dat.bonett2010)
res
predict(res, transf=transf.iabt)

### examine whether women and mixed samples yield different alphas (using raw alpha values)
res <- rma(measure="ARAW", mods = ~ caregivers, ai=ai, mi=mi, ni=ni, data=dat.bonett2010)
res
predict(res, newmods=c(0,1), digits=2)
```

`dat.colditz1994` *Studies on the Effectiveness of the BCG Vaccine for Preventing Tuberculosis*

Description

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

Usage

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

Format

The data frame contains the following columns:

trial	`numeric`	trial number
author	`character`	author(s)
year	`numeric`	publication year
tpos	`numeric`	number of TB positive cases in the treated (vaccinated) group
tneg	`numeric`	number of TB negative cases in the treated (vaccinated) group
cpos	`numeric`	number of TB positive cases in the control (non-vaccinated) group
cneg	`numeric`	number of TB negative cases in the control (non-vaccinated) group
ablat	`numeric`	absolute latitude of the study location (in degrees)
alloc	`character`	method of treatment allocation (random, alternate, or systematic assignment)

Details

The 13 studies provide data in terms of 2x2 tables in the form:

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

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

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

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

Examples

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

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

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

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

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

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

### note: the interpretation of the results is difficult because absolute
### latitude and publication year are strongly correlated (the more recent
### studies were conducted closer to the equator)
plot(dat$ablat, dat$year)
cor(dat$ablat, dat$year)
```

---

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

Description

Results from studies examining the presence of persistent or recurrent bleedings in patients receiving either a histamine H2 antagonist or placebo.

Usage

```
dat.collins1985a
```

Format

The data frame contains the following columns:

id	numeric	study number
ref	numeric	reference number
year	numeric	year of publication
nti	numeric	number of patients in treatment group
xti	numeric	number of patients in treatment group with persistent or recurrent bleedings
nci	numeric	number of patients in placebo group
xci	numeric	number of patients in placebo group with persistent or recurrent bleedings

Details

The data contained in this dataset were obtained from Table I in van Houwelingen, Zwinderman, and Stijnen (1993).

Source

Collins, R., & Langman, M. (1985). Treatment with histamine H2 antagonists in acute upper gastrointestinal hemorrhage. *New England Journal of Medicine*, **313**, 660–666.

van Houwelingen, H. C., Zwinderman, K. H., & Stijnen, T. (1993). A bivariate approach to meta-analysis. *Statistics in Medicine*, **12**, 2273–2284.

References

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

Examples

```
### load data
data(dat.collins1985a)

### calculate (log) odds ratio and sampling variance
dat <- escalc(measure="OR", ai=xci, n1i=nci, ci=xti, n2i=nti,
              data=dat.collins1985a, to="all")
summary(dat, digits=2, transf=exp)

### meta-analysis of log odds ratios using Peto's method
res <- rma.peto(ai=xci, n1i=nci, ci=xti, n2i=nti, data=dat)
summary(res)

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

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

### meta-analysis of log odds ratios using conditional logistic regression model
res <- rma.glmm(measure="OR", ai=xci, n1i=nci, ci=xti, n2i=nti,
```

```
                data=dat, model="CM.EL", method="ML")
summary(res)

## End(Not run)
```

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

Description

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

Usage

```
dat.collins1985b
```

Format

The data frame contains the following columns:

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

Details

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

Source

Collins, R., Yusuf, S., & Peto, R. (1985). Overview of randomised trials of diuretics in pregnancy. *British Medical Journal*, **290**, 17–23.

Examples

```
### load data
data(dat.collins1985b)

### calculate (log) odds ratio and sampling variance
dat <- escalc(measure="OR", n1i=pre.nti, n2i=pre.nci, ai=pre.xti, ci=pre.xci, data=dat.collins1985b)
summary(dat, digits=2, transf=exp)

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

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

### meta-analyses of mortality outcomes (perinatal deaths, stillbirths, and neonatal deaths)
rma.peto(n1i=fup.nti, n2i=fup.nci, ai=ped.xti, ci=ped.xci, data=dat, digits=2, add=0)
rma.peto(n1i=fup.nti, n2i=fup.nci, ai=stb.xti, ci=stb.xci, data=dat, digits=2, add=0)
rma.peto(n1i=fup.nti, n2i=fup.nci, ai=ned.xti, ci=ned.xci, data=dat, digits=2, add=0)
```

---

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

Description

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

Usage

```
dat.curtis1998
```

Format

The data frame contains the following columns:

id	numeric	observation number
paper	numeric	paper number
units	numeric	paper number
genus	character	genus name
species	character	species name
function	character	plant functional group
co2.ambi	numeric	ambient CO2 level (control group)
co2.elev	numeric	elevated CO2 level (treatment group)
units	character	units for CO2 exposure levels
time	numeric	maximum length of time (days) of CO2 exposure
pot	character	growing method (see below)
method	character	CO2 exposure facility (see below)
stock	character	planting stock code
xtrt	character	interacting treatment code (see below)

level	character	interacting treatment level codes (see below)
m1i	numeric	mean plant mass under elevated CO2 level (treatment group)
sd1i	numeric	standard deviation of plant mass underelevated CO2 level (treatment group)
n1i	numeric	number of observations under elevated CO2 level (treatment group)
m2i	numeric	mean plant mass under ambient CO2 level (control group)
sd2i	numeric	standard deviation of plant mass under ambient CO2 level (control group)
n2i	numeric	number of observations under ambient CO2 level (control group)

Details

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

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

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

Source

Hedges, L. V., Gurevitch, J., & Curtis, P. S. (1999). The meta-analysis of response ratios in experimental ecology. *Ecology*, **80**, 1150–1156. (data obtained from *Ecological Archives*, E080-008-S1, at: http://www.esapubs.org/archive/ecol/E080/008/)

References

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

Examples

```
### load data
data(dat.curtis1998)

### calculate log ratio of means and corresponding sampling variances
dat <- escalc(measure="ROM", m1i=m1i, sd1i=sd1i, n1i=n1i, m2i=m2i,
              sd2i=sd2i, n2i=n2i, data=dat.curtis1998)
dat

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

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

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

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

---

dat.hackshaw1998 *Studies on Lung Cancer Risk from ETS Exposure*

Description

Results from 37 studies on the risk of lung cancer from environmental tobacco smoke (ETS) exposure.

Usage

```
dat.hackshaw1998
```

Format

The data frame contains the following columns:

study	numeric	study number
author	character	first author of study
year	numeric	publication year
country	character	country where study was conducted
design	character	either cohort or case-control study
cases	numeric	number of lung cancer cases
yi	numeric	log odds ratio
vi	numeric	corresponding sampling variance

Details

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

Source

Hackshaw, A. K. (1998). Lung cancer and passive smoking. *Statistical Methods in Medical Research*, **7**, 119–136.

Examples

```
### load data
data(dat.hackshaw1998)

### random-effects model
res <- rma(yi, vi, data=dat.hackshaw1998)
res

### obtain predicted average odds ratio
predict(res, transf=exp, digits=2)
```

---

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

Description

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

Usage

```
dat.hart1999
```

Format

The data frame contains the following columns:

trial	numeric	trial number
study	character	study name (abbreviated)
year	numeric	publication year
x1i	numeric	number of strokes in the warfarin group
n1i	numeric	number of patients in the warfarin group
t1i	numeric	total person-time (in years) in the warfarin group
x2i	numeric	number of strokes in the placebo/control group
n2i	numeric	number of patients in the placebo/control group
t2i	numeric	total person-time (in years) in the placebo/control group

compgrp	character	type of comparison group (placebo or control)
prevtype	character	type of prevention (primary or secondary)
trinr	character	target range for the international normalized ratio (INR)

Details

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

Source

Hart, R. G., Benavente, O., McBride, R., & Pearce, L. A. (1999). Antithrombotic therapy to prevent stroke in patients with atrial fibrillation: A meta-analysis. *Annals of Internal Medicine*, **131**, 492–501.

Examples

```
### load data
data(dat.hart1999)

### calculate log incidence rate ratios and corresponding sampling variances
dat <- escalc(measure="IRR", x1i=x1i, x2i=x2i, t1i=t1i, t2i=t2i, data=dat.hart1999)
dat

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

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

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

### meta-analysis of incidence rate differences using a random-effects model
res <- rma(measure="IRD", x1i=x1i, x2i=x2i, t1i=t1i, t2i=t2i, data=dat.hart1999)
res
```

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

Description

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

Usage

```
dat.hine1989
```

Format

The data frame contains the following columns:

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

Details

Hine et al. (1989) conducted a meta-analysis of death rates in randomized controlled trials in which prophylactic lidocaine was administered to patients with confirmed or suspected acute myocardial infarction. The dataset describes the mortality at the end of the assigned treatment period for control and intravenous lidocaine treatment groups for six studies. The question of interest is whether there is a detrimental effect of lidocaine. Because the studies were conducted to compare rates of arrhythmias following a heart attack, the studies, taken individually, are too small to detect important differences in mortality rates.

The data contained in this dataset were obtained from Table I in Normand (1999, p. 322).

Source

Hine, L. K., Laird, N., Hewitt, P., & Chalmers, T. C. (1989). Meta-analytic evidence against prophylactic use of lidocaine in acute myocardial infarction. *Archives of Internal Medicine*, **149**, 2694–2698.

Normand, S. T. (1999). Meta-analysis: Formulating, evaluating, combining, and reporting. *Statistics in Medicine*, **18**, 321–359.

Examples

```
### load data
data(dat.hine1989)
```

```
### calculate risk differences and corresponding sampling variances
dat <- escalc(measure="RD", n1i=n1i, n2i=n2i, ai=ai, ci=ci, data=dat.hine1989)
dat

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

`dat.mcdaniel1994` *Studies on the Validity of Employment Interviews*

Description

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

Usage

```
dat.mcdaniel1994
```

Format

The data frame contains the following columns:

study	`numeric`	study number
ni	`numeric`	sample size of the study
ri	`numeric`	observed correlation
type	`character`	interview type (j = job-related, s = situational, p = psychological)
struct	`character`	interview structure (u = unstructured, s = structured)

Details

The 160 studies provide data in terms of the correlation between employment interview performance and actual job performance. In addition, the interview type and the interview structure are indicated.

McDaniel et al. (1994) describe the interview type and structure variables as follows. "Questions in situational interviews [...] focus on the individual's ability to project what his or her behavior would be in a given situation. [...] Job-related interviews are those in which the interviewer is a personnel officer or hiring authority and the questions attempt to assess past behaviors and job-related information, but most questions are not considered situational. Psychological interviews are conducted by a psychologist, and the questions are intended to assess personal traits, such as dependability." In structured interviews, "the questions and acceptable responses were specified in advance and the responses were rated for appropriateness of content. [...] Unstructured interviews gather applicant information in a less systematic manner than do structured interviews. Although the questions may be specified in advance, they usually are not, and there is seldom a formalized scoring guide. Also, all persons being interviewed are not typically asked the same questions."

The goal of the meta-analysis was to examine the overall criterion-related validity of employment

interviews and to examine whether the validity depends on the type and structure of the interview.

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

Source

McDaniel, M. A., Whetzel, D. L., Schmidt, F. L., & Maurer, S. D. (1994). The validity of employment interviews: A comprehensive review and meta-analysis. *Journal of Applied Psychology*, **79**, 599–616.

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

Examples

```
### load data
data(dat.mcdaniel1994)

### calculate r-to-z transformed correlations and corresponding sampling variances
dat <- escalc(ri=ri, ni=ni, measure="ZCOR", data=dat.mcdaniel1994)
dat

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

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

### mixed-effects model with interview type as factor
### note: job-related interviews is the reference level
rma(yi, vi, mods=~factor(type), data=dat)

### mixed-effects model with interview structure as factor
### note: structured interviews is the reference level
rma(yi, vi, mods=~factor(struct), data=dat)

### note: the interpretation of the results is difficult since all
### situational interviews were structured, almost all psychological
### interviews were unstructured, and actually for the majority of
### the psychological interviews it was unknown whether the interview
### was structured or unstructured
table(dat$type, dat$struct, useNA="always")

### meta-analysis of raw correlations using a random-effects model
res <- rma(ri=ri, ni=ni, measure="COR", data=dat.mcdaniel1994)
res
```

dat.normand1999 *Studies on the Length of Hospital Stay of Stroke Patients*

Description

Results from 9 studies on the length of the hospital stay of stroke patients under specialized care and under conventional/routine (non-specialist) care.

Usage

```
dat.normand1999
```

Format

The data frame contains the following columns:

study	numeric	study number
source	character	source of data
n1i	numeric	number of patients under specialized care
m1i	numeric	mean length of stay (in days) under specialized care
sd1i	numeric	standard deviation of the length of stay under specialized care
n2i	numeric	number of patients under routine care
m2i	numeric	mean length of stay (in days) under routine care
sd2i	numeric	standard deviation of the length of stay under routine care

Details

The 9 studies provide data in terms of the mean length of the hospital stay (in days) of stroke patients under specialized care and under conventional/routine (non-specialist) care. The goal of the meta-analysis was to examine the hypothesis whether specialist stroke unit care will result in a shorter length of hospitalization compared to routine management.

Source

Normand, S. T. (1999). Meta-analysis: Formulating, evaluating, combining, and reporting. *Statistics in Medicine*, **18**, 321–359.

Examples

```
### load data
data(dat.normand1999)

### calculate mean differences and corresponding sampling variances
dat <- escalc(measure="MD", m1i=m1i, sd1i=sd1i, n1i=n1i, m2i=m2i,
              sd2i=sd2i, n2i=n2i, data=dat.normand1999)
dat

### meta-analysis of mean differences using a random-effects model
```

```
res <- rma(yi, vi, data=dat)
res

### meta-analysis of standardized mean differences using a random-effects model
res <- rma(m1i=m1i, sd1i=sd1i, n1i=n1i, m2i=m2i, sd2i=sd2i, n2i=n2i,
           measure="SMD", data=dat.normand1999, slab=source)
res

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

`dat.pritz1997` *Studies on the Effectiveness of Hyperdynamic Therapy for Treating Cerebral Vasospasm*

Description

Results from 14 studies on the effectiveness of hyperdynamic therapy for treating cerebral vasospasm.

Usage

```
dat.pritz1997
```

Format

The data frame contains the following columns:

study	`numeric`	study number
authors	`character`	study authors
xi	`numeric`	number of patients that improved with hyperdynamic therapy
ni	`numeric`	total number of patients treated

Details

As described in Zhou et al. (1999), "hyperdynamic therapy refers to induced hypertension and hypervolaemia (volume expansion) to treat ischaemic symptoms due to vasospasm, and the success of this therapy is defined as clinical improvement in terms of neurologic deficits." For each study that was included in the meta-analysis, the dataset includes information on the number of patients that improved under this form of therapy and the total number of patients that were treated. The goal of the meta-analysis is to estimate the true (average) success rate of hyperdynamic therapy.

Source

Zhou, X.-H., Brizendine, E. J., & Pritz, M. B. (1999). Methods for combining rates from several studies. *Statistics in Medicine*, **18**, 557–566.

References

Pritz, M. B. (1997). Treatment of cerebral vasospasm due to aneurysmal subarachnoid hemorrhage: Past, present, and future of hyperdynamic therapy. *Neurosurgery Quarterly*, **7**, 273–285.

Examples

```
### load data
data(dat.pritz1997)

### computation of "weighted average" in Zhou et al. (1999), Table IV
dat <- escalc(measure="PR", xi=xi, ni=ni, data=dat.pritz1997, add=0)
theta.hat    <- sum(dat$ni * dat$yi) / sum(dat$ni)
se.theta.hat <- sqrt(sum(dat$ni^2 * dat$vi) / sum(dat$ni)^2)
ci.lb        <- theta.hat - 1.96*se.theta.hat
ci.ub        <- theta.hat + 1.96*se.theta.hat
round(c(estimate = theta.hat, se = se.theta.hat, ci.lb = ci.lb, ci.ub = ci.ub), 4)

### random-effects model with raw proportions
dat <- escalc(measure="PR", xi=xi, ni=ni, data=dat.pritz1997)
res <- rma(yi, vi, method="REML", data=dat)
predict(res)

### random-effects model with logit transformed proportions
dat <- escalc(measure="PLO", xi=xi, ni=ni, data=dat.pritz1997)
res <- rma(yi, vi, method="REML", data=dat)
predict(res, transf=transf.ilogit)
```

---

dat.raudenbush1985 *Studies on Assessing the Effects of Teacher Expectations on Pupil IQ*

Description

Results from 19 studies examining how teachers' expectations about their pupils can influence actual IQ levels.

Usage

```
dat.raudenbush1985
```

Format

The data frame contains the following columns:

study	numeric	study number
author	character	study author(s)
year	numeric	publication year
weeks	numeric	weeks of contact prior to expectancy induction
setting	character	whether tests were group or individually administered
tester	character	whether test administrator was aware or blind

yi	numeric	standardized mean difference
vi	numeric	corresponding sampling variance

Details

In the so-called 'Pygmalion study' (Rosenthal & Jacobson, 1968), all of the children in an elementary school were administered a test labeled the 'Harvard Test of Inflected Acquisition'. After explaining that this newly designed instrument had identified those children most likely to show dramatic intellectual growth during the coming year, the experimenters gave the names of these 'bloomers' to the teachers. In truth, the test was a traditional IQ test and the 'bloomers' were a randomly selected 20

In the following years, a series of studies were conducted attempting to replicate this rather controversial finding. However, the great majority of those studies were unable to demonstrate a statistically significant difference between the two experimental groups in terms of IQ scores. Raudenbush (1984) conducted a meta-analysis based on 19 such studies to further examine the evidence for the existence of the 'Pygmalion effect'. The dataset includes the results from these studies.

The effect size measure used for the meta-analysis was the standardized mean difference (`yi`), with positive values indicating that the supposed 'bloomers' had, on average, higher IQ scores than those in the control group. The `weeks` variable indicates the number of weeks of prior contact between teachers and students before the expectancy induction. Testing was done either in a group setting or individually, which is indicated by the `setting` variable. Finally, the `tester` variable indicates whether the test administrators were either aware or blind to the researcher-provided designations of the children's intellectual potential.

The data contained in this dataset were obtained from Raudenbush & Bryk (1985) with information on the `setting` and `tester` variables extracted from Raudenbush (1984).

Source

Raudenbush, S. W. (1984). Magnitude of teacher expectancy effects on pupil IQ as a function of the credibility of expectancy induction: A synthesis of findings from 18 experiments. *Journal of Educational Psychology*, **76**, 85–97.

Raudenbush, S. W., & Bryk, A. S. (1985). Empirical Bayes meta-analysis. *Journal of Educational Statistics*, **10**, 75–98.

Examples

```
### load data
data(dat.raudenbush1985)

### random-effects model
res <- rma(yi, vi, data=dat.raudenbush1985)
res
```

escalc *Calculate Effect Size and Outcome Measures*

Description

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

Usage

```
escalc(measure, formula, ...)

## Default S3 method:
escalc(measure, formula, ai, bi, ci, di, n1i, n2i, x1i, x2i, t1i, t2i,
       m1i, m2i, sd1i, sd2i, xi, mi, ri, ti, sdi, ni, data, slab, subset,
       add=1/2, to="only0", drop00=FALSE, vtype="LS",
       var.names=c("yi","vi"), append=TRUE, replace=TRUE, digits=4, ...)

## S3 method for class 'formula'
escalc(measure, formula, weights, data,
       add=1/2, to="only0", drop00=FALSE, vtype="LS",
       var.names=c("yi","vi"), digits=4, ...)
```

Arguments

measure	a character string indicating which effect size or outcome measure should be calculated. See 'Details' for possible options and how the data should be specified.
formula	when using the formula interface of the function (see 'Details' below), a model formula specifying the data structure should be specified via this argument. When not using the formula interface, this argument should be ignored and the data required to calculate the effect sizes or outcomes are then passed to the function via the following set of arguments. See 'Details'.
weights	vector of weights to specify the group sizes or cell frequencies (only needed when using the formula interface). See 'Details'.
ai	vector to specify the 2x2 table frequencies (upper left cell).
bi	vector to specify the 2x2 table frequencies (upper right cell).
ci	vector to specify the 2x2 table frequencies (lower left cell).
di	vector to specify the 2x2 table frequencies (lower right cell).
n1i	vector to specify the group sizes or row totals (first group/row).
n2i	vector to specify the group sizes or row totals (second group/row).
x1i	vector to specify the number of events (first group).
x2i	vector to specify the number of events (second group).
t1i	vector to specify the total person-times (first group).

t2i	vector to specify the total person-times (second group).
m1i	vector to specify the means (first group or time point).
m2i	vector to specify the means (second group or time point).
sd1i	vector to specify the standard deviations (first group or time point).
sd2i	vector to specify the standard deviations (second group or time point).
xi	vector to specify the frequencies of the event of interest.
mi	vector to specify the frequencies of the complement of the event of interest or the group means.
ri	vector to specify the raw correlation coefficients.
ti	vector to specify the total person-times.
sdi	vector to specify the standard deviations.
ni	vector to specify the sample/group sizes.
data	optional data frame containing the variables given to the arguments above.
slab	optional vector with unique labels for the studies.
subset	optional vector indicating the subset of studies that should be used. This can be a logical vector or a numeric vector indicating the indices of the studies to include.
add	a non-negative number indicating the amount to add to zero cells, counts, or frequencies. See 'Details'.
to	a character string indicating when the values under add should be added (either "all", "only0", "if0all", or "none"). See 'Details'.
drop00	logical indicating whether studies with no cases/events (or only cases) in both groups should be dropped when calculating the observed outcomes of the individual studies. See 'Details'.
vtype	a character string indicating the type of sampling variances to calculate (either "LS", "UB", "ST", or vtype="CS"). See 'Details'.
var.names	a character string with two elements, specifying the name of the variable for the observed outcomes and the name of the variable for the corresponding sampling variances (default is "yi" and "vi").
append	logical indicating whether the data frame specified via the data argument (if one has been specified) should be returned together with the observed outcomes and corresponding sampling variances (default is TRUE).
replace	logical indicating whether existing values for yi and vi in the data frame should be replaced or not. Only relevant when append=TRUE and the data frame already contains the yi and vi variables. If replace=TRUE (the default), all of the existing values will be overwritten. If replace=FALSE, only NA values will be replaced. See 'Value' section below for more details.
digits	integer specifying the number of decimal places to which the printed results should be rounded (default is 4). Note that the values are stored without rounding in the returned object.
...	other arguments.

Details

Before a meta-analysis can be conducted, the relevant results from each study must be quantified in such a way that the resulting values can be further aggregated and compared. Depending on (a) the goals of the meta-analysis, (b) the design and types of studies included, and (c) the information provided therein, one of the various effect size or outcome measures described below may be appropriate for the meta-analysis and can be computed with the `escalc` function.

The `measure` argument is a character string specifying which outcome measure should be calculated (see below for the various options), arguments `ai` through `ni` are then used to specify the information needed to calculate the various measures (depending on the chosen outcome measure, different arguments need to be specified), and `data` can be used to specify a data frame containing the variables given to the previous arguments. The `add`, `to`, and `drop00` arguments may be needed when dealing with frequency or count data that may need special handling when some of the frequencies or counts are equal to zero (see below for details). Finally, the `vtype` argument is used to specify how to estimate the sampling variances (again, see below for details).

To provide a structure to the various effect size or outcome measures that can be calculated with the `escalc` function, we can distinguish between measures that are used to:

- contrast two (either experimentally created or naturally occuring) groups,
- describe the direction and strength of the association between two variables,
- summarize some characteristic or attribute of individual groups.

Furthermore, where appropriate, we can further distinguish between measures that are applicable when the characteristic, response, or dependent variable assessed in the individual studies is:

- a dichotomous (binary) variable (e.g., remission versus no remission),
- a count of events per time unit (e.g., number of migraines per year),
- a quantitative variable (e.g., amount of depression as assessed by a rating scale).

Outcome Measures for Two-Group Comparisons:

In many meta-analyses, the goal is to synthesize the results from studies that compare or contrast two groups. The groups may be experimentally defined (e.g., a treatment and a control group created via random assignment) or may naturally occur (e.g., men and women, employees working under high- versus low-stress conditions, people exposed to some environmental risk factor versus those not exposed).

Measures for Dichotomous Variables:

In various fields (such as the health and medical sciences), the response variable measured is often dichotomous (binary), so that the data from a study comparing two different groups can be expressed in terms of a 2x2 table, such as:

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

where `ai`, `bi`, `ci`, and `di` denote the cell frequencies (i.e., the number of people falling into a particular category) and `n1i` and `n2i` the row totals (i.e., the group sizes).

For example, in a set of randomized clinical trials, group 1 and group 2 may refer to the treatment and placebo/control group, respectively, with outcome 1 denoting some event of interest (e.g.,

death, complications, failure to improve under the treatment) and outcome 2 its complement. Similarly, in a set of cohort studies, group 1 and group 2 may denote those who engage in and those who do not engage in a potentially harmful behavior (e.g., smoking), with outcome 1 denoting the development of a particular disease (e.g., lung cancer) during the follow-up period. Finally, in a set of case-control studies, group 1 and group 2 may refer to those with the disease (i.e., cases) and those free of the disease (i.e., controls), with outcome 1 denoting, for example, exposure to some risk environmental risk factor and outcome 2 non-exposure. Note that in all of these examples, the stratified sampling scheme fixes the row totals (i.e., the group sizes) by design.

A meta-analysis of studies reporting results in terms of 2x2 tables can be based on one of several different outcome measures, including the relative risk (risk ratio), the odds ratio, the risk difference, and the arcsine transformed risk difference (e.g., Fleiss & Berlin, 2009, Ruecker et al., 2009). For any of these outcome measures, one needs to specify the cell frequencies via the `ai`, `bi`, `ci`, and `di` arguments (or alternatively, one can use the `ai`, `ci`, `n1i`, and `n2i` arguments). The options for the `measure` argument are then:

- `"RR"` for the *log relative risk*.
- `"OR"` for the *log odds ratio*.
- `"RD"` for the *risk difference*.
- `"AS"` for the *arcsine transformed risk difference* (Ruecker et al., 2009).
- `"PETO"` for the *log odds ratio* estimated with Peto's method (Yusuf et al., 1985).

Note that the log is taken of the relative risk and the odds ratio, which makes these outcome measures symmetric around 0 and yields corresponding sampling distributions that are closer to normality.

Cell entries with a zero count can be problematic, especially for the relative risk and the odds ratio. Adding a small constant to the cells of the 2x2 tables is a common solution to this problem. When `to="only0"` (the default), the value of `add` (the default is 1/2) is added to each cell of those 2x2 tables with at least one cell equal to 0. When `to="all"`, the value of `add` is added to each cell of all 2x2 tables. When `to="if0all"`, the value of `add` is added to each cell of all 2x2 tables, but only when there is at least one 2x2 table with a zero cell. Setting `to="none"` or `add=0` has the same effect: No adjustment to the observed table frequencies is made. Depending on the outcome measure and the data, this may lead to division by zero inside of the function (when this occurs, the resulting value is recoded to `NA`). Also, studies where `ai=ci=0` or `bi=di=0` may be considered to be uninformative about the size of the effect and dropping such studies has sometimes been recommended (Higgins & Green, 2008). This can be done by setting `drop00=TRUE`. The counts for such studies will be then be set to `NA`.

A dataset corresponding to data of this type is provided in `dat.bcg`.

Assuming that the dichotomous outcome is actually a dichotomized version of the responses on an underlying quantitative scale, it is also possible to estimate the standardized mean difference based on 2x2 table data, using either the probit transformed risk difference or a transformation of the odds ratio (e.g., Chinn, 2000; Hasselblad & Hedges, 1995; Sanchez-Meca et al., 2003). The options for the `measure` argument are then:

- `"PBIT"` for the *probit transformed risk difference* as an estimate of the standardized mean difference.
- `"OR2D"` for *transformed odds ratio* as an estimate of the standardized mean difference.

The probit transformation assumes that the responses on the underlying quantitative scale are normally distributed, while the odds ratio transformation assumes that the responses follow logistic distributions within each group.

Measures for Event Counts:
In medical and epidemiological studies comparing two different groups (e.g., treated versus untreated patients, exposed versus unexposed individuals), results are sometimes reported in terms of event counts (i.e., the number of events, such as strokes or myocardial infarctions) over a certain period of time. In particular, assume that the studies report data in the form:

	number of events	total person-time
group 1	`x1i`	`t1i`
group 2	`x2i`	`t2i`

where `x1i` and `x2i` denote the total number of events in the first and the second group, respectively, and `t1i` and `t2i` the corresponding total person-times at risk. Often, the person-time is measured in years, so that `t1i` and `t2i` denote the total number of follow-up years in the two groups.
Note that this form of data is fundamentally different from that described in the previous section, since the total follow-up time may differ even for groups of the same size and the individuals studied may experience the event of interest multiple times. Hence, different outcome measures than the ones described in the previous section must be considered when data are reported in this format. These inlude the incidence rate ratio, the incidence rate difference, and the square-root transformed incidence rate difference (Bagos & Nikolopoulos, 2009). For any of these outcome measures, one needs to specify the total number of events via the `x1i` and `x2i` arguments and the corresponding total person-times via the `t1i` and `t2i` arguments.
The options for the `measure` argument are then:

- `"IRR"` for the *log incidence rate ratio*.
- `"IRD"` for the *incidence rate difference*.
- `"IRSD"` for the *square-root transformed incidence rate difference*.

Note that the log is taken of the incidence rate ratio, which makes this outcome measure symmetric around 0 and yields a corresponding sampling distribution that is closer to normality.
Studies with zero events in one or both groups can be problematic, especially for the incidence rate ratio. Adding a small constant to the number of events is a common solution to this problem. When `to="only0"` (the default), the value of `add` (the default is 1/2) is added to `x1i` and `x2i` only in the studies that have zero events in one or both groups. When `to="all"`, the value of `add` is added to `x1i` and `x2i` in all studies. When `to="if0all"`, the value of `add` is added to `x1i` and `x2i` in all studies, but only when there is at least one study with zero events in one or both groups. Setting `to="none"` or `add=0` has the same effect: No adjustment to the observed number of events is made. Depending on the outcome measure and the data, this may lead to division by zero inside of the function (when this occurs, the resulting value is recoded to `NA`).
Like for 2x2 table data, studies where `x1i=x2i=0` may be considered to be uninformative about the size of the effect and dropping such studies has sometimes been recommended. This can be done by setting `drop00=TRUE`. The counts for such studies will be then be set to `NA`.
A dataset corresponding to data of this type is provided in `dat.hart1999`.

Measures for Quantitative Variables:
When the response or dependent variable assessed in the individual studies is measured on some quantitative scale, it is customary to report certain summary statistics, such as the mean and standard deviation of the scores. The data layout for a study comparing two groups with respect to such a variable is then of the form:

	mean	standard deviation	group size

group 1	m1i	sd1i	n1i
group 2	m2i	sd2i	n2i

where m1i and m2i are the observed means of the two groups, sd1i and sd2i the observed standard deviations, and n1i and n2i the number of individuals in each group. Again, the two groups may be experimentally created (e.g., a treatment and control group based on random assignment) or naturally occurring (e.g., men and women). In either case, the raw mean difference, the standardized mean difference, and the ratio of means (also called response ratio) are useful outcome measures when meta-analyzing studies of this type (e.g., Borenstein, 2009). In addition, the (log) odds ratio can be estimated based on data of this type, using a simple transformation of the standardized mean difference (e.g., Chinn, 2000; Hasselblad & Hedges, 1995).

The options for the measure argument are then:

- "MD" for the *raw mean difference*.
- "SMD" for the *standardized mean difference*.
- "SMDH" for the *standardized mean difference* without assuming equal population variances in the two groups (Bonett, 2008, 2009).
- "ROM" for the *log transformed ratio of means* (Hedges et al., 1999).
- "D2OR" for the *transformed standardized mean difference* as an estimate of the log odds ratio.

Note that the log is taken of the ratio of means, which makes this outcome measures symmetric around 0 and yields a corresponding sampling distribution that is closer to normality (however, note that if m1i and m2i have opposite signs, this outcome measure cannot be computed).

The negative bias in the standardized mean difference is automatically corrected for within the function, yielding Hedges' g for measure="SMD" (Hedges, 1981). Similarly, the same bias correction is applied for measure="SMDH" (Bonett, 2009). Finally, for measure="SMD", one can choose between vtype="LS" (the default) and vtype="UB". The former uses a large sample approximation to compute the sampling variances. The latter provides unbiased estimates of the sampling variances.

A dataset corresponding to data of this type is provided in dat.normand1999 (for mean differences and standardized mean differences). A dataset showing the use of the ratio of means measure is provided in dat.curtis1998.

Outcome Measures for Variable Association:

Meta-analyses are often used to synthesize studies that examine the direction and strength of the association between two variables measured concurrently and/or without manipulation by experimenters. In this section, a variety of outcome measures will be discussed that may be suitable for a meta-analyses with this purpose. We can distinguish between measures that are applicable when both variables are measured on quantitative scales, when both variables measured are dichotomous, and when the two variables are of mixed types.

Measures for Two Quantitative Variables:

The (Pearson or product moment) correlation coefficient quantifies the direction and strength of the (linear) relationship between two quantitative variables and is therefore frequently used as the outcome measure for meta-analyses (e.g., Borenstein, 2009). Two alternative measures are a bias-corrected version of the correlation coefficient and Fisher's r-to-z transformed coefficient.

For these measures, one needs to specify ri, the vector with the raw correlation coefficients, and ni, the corresponding sample sizes. The options for the measure argument are then:

- `"COR"` for the *raw correlation coefficient*.
- `"UCOR"` for the *raw correlation coefficient* corrected for its slight negative bias (based on equation 2.7 in Olkin & Pratt, 1958).
- `"ZCOR"` for the *Fisher's r-to-z transformed correlation coefficient* (Fisher, 1921).

For `measure="COR"` and `measure="UCOR"`, one can choose between `vtype="LS"` (the default) and `vtype="UB"`. The former uses a large sample approximation to compute the sampling variances. The latter provides approximately unbiased estimates of the sampling variances (see Hedges, 1989).
A dataset corresponding to data of this type is provided in `dat.mcdaniel1994`.

Measures for Two Dichotomous Variables:
When the goal of a meta-analysis is to examine the relationship between two dichotomous variables, the data for each study can again be presented in the form of a 2x2 table, except that there may not be a clear distinction between the group (i.e., the row) and the outcome (i.e., the column) variable. Moreover, the table may be a result of cross-sectional (i.e., multinomial) sampling, where none of the table margins (except the total sample size) is fixed by the study design.
The phi coefficient and the odds ratio are commonly used measures of association for 2x2 table data (e.g., Fleiss & Berlin, 2009). The latter is particularly advantageous, as it is directly comparable to values obtained from stratified sampling (as described earlier). Yule's Q and Yule's Y (Yule, 1912) are additional measures of association for 2x2 table data (although they are not typically used in meta-analyses). Finally, assuming that the two dichotomous variables are actually dichotomized versions of the responses on two underlying quantitative scales (and assuming that the two variables follow a bivariate normal distribution), it is also possible to estimate the correlation between the two variables using the tetrachoric correlation coefficient (Pearson, 1900; Kirk, 1973).
For any of these outcome measures, one needs to specify the cell frequencies via the `ai`, `bi`, `ci`, and `di` arguments. The options for the `measure` argument are then:

- `"OR"` for the *log odds ratio*.
- `"PHI"` for the *phi coefficient*.
- `"YUQ"` for *Yule's Q* (Yule, 1912).
- `"YUY"` for *Yule's Y* (Yule, 1912).
- `"RTET"` for the *tetrachoric correlation*.

Tables with one or two zero counts are handled as described earlier.

Measures for Mixed Variable Types:
Finally, we will consider outcome measures that can be used to describe the relationship between two variables, where one variable is dichotomous and the other variable measures some quantitative characteristic. In that case, it is likely that study authors again report summary statistics, such as the mean and standard deviation of the scores within the two groups (defined by the dichotomous variable). In that case, one can compute the point-biserial (Tate, 1954) as a measure of association between the two variables. If the dichotomous variable is actually a dichotomized version of the responses on an underlying quantitative scale (and assuming that the two variables follow a bivariate normal distribution), it is also possible to estimate the correlation between the two variables using the biserial correlation coefficient (Pearson, 1909; Soper, 1914).
Here, one again needs to specify `m1i` and `m2i` for the observed means of the two groups, `sd1i` and `sd2i` for the observed standard deviations, and `n1i` and `n2i` for the number of individuals in each group. The options for the `measure` argument are then:

- "RPB" for the *point-biserial correlation*.
- "RBIS" for the *biserial correlation*.

For measure="RPB", one must indicate via vtype="ST" or vtype="CS" whether the data for the studies were obtained using stratified or cross-sectional (i.e., multinomial) sampling, respectively (it is also possible to specify an entire vector for the vtype argument in case the sampling schemes differed for the various studies).

Outcome Measures for Individual Groups:

In this section, outcome measures will be described which may be useful when the goal of a meta-analysis is to synthesize studies that characterize some property of individual groups. We will again distinguish between measures that are applicable when the characteristic of interest is a dichotomous variable, when the characteristic represents an event count, or when the characteristic assessed is a quantiative variable.

Measures for Dichotomous Variables:

A meta-analysis may be conducted to aggregate studies that provide data for individual groups with respect to a dichotomous dependent variable. Here, one needs to specify xi and ni, denoting the number of individuals experiencing the event of interest and the total number of individuals, respectively. Instead of specifying ni, one can use mi to specify the number of individuals that do not experience the event of interest. The options for the measure argument are then:

- "PR" for the *raw proportion*.
- "PLN" for the *log transformed proportion*.
- "PLO" for the *logit transformed proportion* (i.e., log odds).
- "PAS" for the *arcsine transformed proportion*.
- "PFT" for the *Freeman-Tukey double arcsine transformed proportion* (Freeman & Tukey, 1950).

Zero cell entries can be problematic for certain outcome measures. When to="only0" (the default), the value of add (the default is 1/2) is added to xi and mi only for studies where xi or mi is equal to 0. When to="all", the value of add is added to xi and mi in all studies. When to="if0all", the value of add is added in all studies, but only when there is at least one study with a zero value for xi or mi. Setting to="none" or add=0 has the same effect: No adjustment to the observed values is made. Depending on the outcome measure and the data, this may lead to division by zero inside of the function (when this occurs, the resulting value is recoded to NA). A dataset corresponding to data of this type is provided in dat.pritz1997.

Measures for Event Counts:

Various measures can be used to characterize individual groups when the dependent variable assessed is an event count. Here, one needs to specify xi and ti, denoting the total number of events that occurred and the total person-time at risk, respectively. The options for the measure argument are then:

- "IR" for the *raw incidence rate*.
- "IRLN" for the *log transformed incidence rate*.
- "IRS" for the *square-root transformed incidence rate*.
- "IRFT" for the *Freeman-Tukey transformed incidence rate* (Freeman & Tukey, 1950).

Studies with zero events can be problematic, especially for the log transformed incidence rate. Adding a small constant to the number of events is a common solution to this problem. When to="only0" (the default), the value of add (the default is 1/2) is added to xi only in the studies

that have zero events. When `to="all"`, the value of `add` is added to `xi` in all studies. When `to="if0all"`, the value of `add` is added to `xi` in all studies, but only when there is at least one study with zero events. Setting `to="none"` or `add=0` has the same effect: No adjustment to the observed number of events is made. Depending on the outcome measure and the data, this may lead to division by zero inside of the function (when this occurs, the resulting value is recoded to `NA`).

Measures for Quantitative Variables:
The goal of a meta-analysis may also be to characterize individual groups, where the response, characteristic, or dependent variable assessed in the individual studies is measured on some quantitative scale. In the simplest case, the raw mean for the quantitative variable is reported for each group, which then becomes the observed outcome for the meta-analysis. Here, one needs to specify `mi`, `sdi`, and `ni` for the observed means, the observed standard deviations, and the sample sizes, respectively. The only option for the `measure` argument is then:

- `"MN"` for the *raw mean*.

A more complicated situation arises when the purpose of the meta-analysis is to assess the amount of change within individual groups. Here, either the raw mean change or standardized versions thereof can be used as outcome measures (Becker, 1988; Gibbons et al., 1993; Morris, 2000). Here, one needs to specify `m1i` and `m2i`, the observed means at the two measurement occasions, `sd1i` and `sd2i` for the corresponding observed standard deviations, `ri` for the correlation between the scores observed at the two measurement occasions, and `ni` for the sample size. The options for the `measure` argument are then:

- `"MC"` for the *raw mean change*.
- `"SMCC"` for the *standardized mean change* using change score standardization.
- `"SMCR"` for the *standardized mean change* using raw score standardization.

See also Morris and DeShon (2002) for a thorough discussion of the difference between the change score measures.

Other Outcome Measures for Meta-Analyses:

Other outcome measures are sometimes used for meta-analyses that do not directly fall into the categories above. These are described in this section.

Cronbach's alpha and Transformations Thereof:
Meta-analytic methods can also be used to aggregate Cronbach's alpha values. This is usually referred to as a 'reliability generalization meta-analysis' (Vacha-Haase, 1998). Here, one needs to specify `ai`, `mi`, and `ni` for the observed alpha values, the number of items/replications/parts of the measurement instrument, and the sample sizes, respectively. One can either directly analyze the raw Cronbach's alpha values or transformations thereof (Bonett, 2002, 2010; Hakstian & Whalen, 1976). The options for the `measure` argument are then:

- `"ARAW"` for *raw alpha* values.
- `"AHW"` for *transformed alpha values* (Hakstian & Whalen, 1976).
- `"ABT"` for *transformed alpha values* (Bonett, 2002).

Note that the transformations implemented here are slightly different from the ones described by Hakstian and Whalen (1976) and Bonett (2002). In particular, for `"AHW"`, the transformation $1 - (1 - \alpha)^{1/3}$ is used, while for `"ABT"`, the transformation $-ln(1 - \alpha)$ is used. This ensures that the transformed values are monotonically increasing functions of alpha.
A dataset corresponding to data of this type is provided in `dat.bonett2010`.

Formula Interface:

There are two general ways of specifying the data for computing the various effect size or outcome measures when using the `escalc` function, the default and a formula interface. When using the default interface, which is described above, the information needed to compute the various outcome measures is passed to the function via the various arguments outlined above (i.e., arguments `ai` through `ni`).

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

Outcome Measures for Two-Group Comparisons:

Measures for Dichotomous Variables:

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

Measures for Event Counts:

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

Measures for Quantitative Variables:

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

Outcome Measures for Variable Association:

Measures for Two Quantitative Variables:

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

Measures for Two Dichotomous Variables:

Here, the data layout is assumed to be the same as for two-group comparisons with dichotomous variables. Hence, the `formula` argument is specified in the same manner.

Measures for Mixed Variable Types:

Here, the data layout is assumed to be the same as for two-group comparisons with quantitative variables. Hence, the `formula` argument is specified in the same manner.

Outcome Measures for Individual Groups:

Measures for Dichotomous Variables:

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

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

Measures for Quantitative Variables:
For this outcome measures, the `formula` argument takes the form `means/sds ~ 1 | study`, where `study` is a factor specifying the study factor. The left-hand side of the formula is composed of two parts, with the first variable for the means and the second variable for the standard deviations. The `weights` argument is used to specify the sample sizes.
Note: The formula interface is (currently) not implemented for the raw mean change and the standardized mean change measures.

Other Outcome Measures for Meta-Analyses:

Cronbach's alpha and Transformations Thereof:
For these outcome measures, the `formula` argument takes the form `alpha/items ~ 1 | study`, where `study` is a factor specifying the study factor. The left-hand side of the formula is composed of two parts, with the first variable for the Cronbach's alpha values and the second variable for the number of items.

Value

An object of class `c("escalc","data.frame")`. The object is a data frame containing the following components:

`yi`	observed outcomes or effect size estimates.
`vi`	corresponding (estimated) sampling variances.

If `append=TRUE` and a data frame was specified via the `data` argument, then `yi` and `vi` are append to this data frame. Note that the `var.names` argument actually specifies the names of these two variables.

If the data frame already contains two variables with names as specified by the `var.names` argument, the values for these two variables will be overwritten when `replace=TRUE` (which is the default). By setting `replace=FALSE`, only values that are `NA` will be replaced.

The object is formated and printed with the `print.escalc` function. The `summary.escalc` function can be used to obtain confidence intervals for the individual outcomes.

Note

The variable names specified under `var.names` should be syntactically valid variable names. If necessary, they are adjusted so that they are.

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

Author(s)

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

References

Bagos, P. G., & Nikolopoulos, G. K. (2009). Mixed-effects Poisson regression models for meta-analysis of follow-up studies with constant or varying durations. *The International Journal of Biostatistics*, **5**(1), article 21.

Becker, B. J. (1988). Synthesizing standardized mean-change measures. *British Journal of Mathematical and Statistical Psychology*, **41**, 257–278.

Bonett, D. G. (2002). Sample size requirements for testing and estimating coefficient alpha. *Journal of Educational and Behavioral Statistics*, **27**, 335–340.

Bonett, D. G. (2008). Confidence intervals for standardized linear contrasts of means. *Psychological Methods*, **13**, 99–109.

Bonett, D. G. (2009). Meta-analytic interval estimation for standardized and unstandardized mean differences. *Psychological Methods*, **14**, 225–238.

Bonett, D. G. (2010). Varying coefficient meta-analytic methods for alpha reliability. *Psychological Methods*, **15**, 368–385.

Borenstein, M. (2009). Effect sizes for continuous data. In H. Cooper, L. V. Hedges, & J. C. Valentine (Eds.), *The handbook of research synthesis and meta-analysis* (2nd ed., pp. 221–235). New York: Russell Sage Foundation.

Chinn, S. (2000). A simple method for converting an odds ratio to effect size for use in meta-analysis. *Statistics in Medicine*, **19**, 3127–3131.

Fisher, R. A. (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron*, **1**, 1–32.

Fleiss, J. L., & Berlin, J. (2009). Effect sizes for dichotomous data. In H. Cooper, L. V. Hedges, & J. C. Valentine (Eds.), *The handbook of research synthesis and meta-analysis* (2nd ed., pp. 237–253). New York: Russell Sage Foundation.

Freeman, M. F., & Tukey, J. W. (1950). Transformations related to the angular and the square root. *Annals of Mathematical Statistics*, **21**, 607–611.

Gibbons, R. D., Hedeker, D. R., & Davis, J. M. (1993). Estimation of effect size from a series of experiments involving paired comparisons. *Journal of Educational Statistics*, **18**, 271–279.

Hakstian, A. R., & Whalen, T. E. (1976). A k-sample significance test for independent alpha coefficients. *Psychometrika*, **41**, 219–231.

Hasselblad, V., & Hedges, L. V. (1995). Meta-analysis of screening and diagnostic tests. Psychological Bulletin, 117(1), 167-178.

Hedges, L. V. (1981). Distribution theory for Glass's estimator of effect size and related estimators. *Journal of Educational Statistics*, **6**, 107–128.

Hedges, L. V. (1989). An unbiased correction for sampling error in validity generalization studies. *Journal of Applied Psychology*, **74**, 469–477.

Hedges, L. V., Gurevitch, J., & Curtis, P. S. (1999). The meta-analysis of response ratios in experimental ecology. *Ecology*, **80**, 1150–1156.

Higgins, J. P. T., & Green, S. (Eds.) (2008). *Cochrane handbook for systematic reviews of interventions*. Chichester, Englang: Wiley.

Kirk, D. B. (1973). On the numerical approximation of the bivariate normal (tetrachoric) correlation coefficient. *Psychometrika*, **38**, 259–268.

Morris, S. B. (2000). Distribution of the standardized mean change effect size for meta-analysis on repeated measures. *British Journal of Mathematical and Statistical Psychology*, **53**, 17–29.

Morris, S. B., & DeShon, R. P. (2002). Combining effect size estimates in meta-analysis with repeated measures and independent-groups designs. *Psychological Methods*, **7**, 105–125.

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.

Sanchez-Meca, J., Marin-Martinez, F., & Chacon-Moscoso, S. (2003). Effect-size indices for dichotomized outcomes in meta-analysis. *Psychological Methods*, **8**, 448–467.

Olkin, I., & Pratt, J. W. (1958). Unbiased estimation of certain correlation coefficients. *Annals of Mathematical Statistics*, **29**, 201–211.

Pearson, K. (1900). Mathematical contribution to the theory of evolution. VII. On the correlation of characters not quantitatively measurable. *Philosophical Transactions of the Royal Society of London, Series A*, **195**, 1–47.

Pearson, K. (1909). On a new method of determining correlation between a measured character A, and a character B, of which only the percentage of cases wherein B exceeds (or falls short of) a given intensity is recorded for each grade of A. *Biometrika*, **7**, 96–105.

Soper, H. E. (1914). On the probable error of the bi-serial expression for the correlation coefficient. *Biometrika*, **10**, 384–390.

Tate, R. F. (1954). Correlation between a discrete and a continuous variable: Point-biserial correlation. *Annals of Mathematical Statistics*, **25**, 603–607.

Vacha-Haase, T. (1998). Reliability generalization: Exploring variance in measurement error affecting score reliability across studies. *Educational and Psychological Measurement*, **58**, 6–20.

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

Yule, G. U. (1912). On the methods of measuring association between two attributes. *Journal of the Royal Statistical Society*, **75**, 579–652.

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

`print.escalc`, `summary.escalc`, `rma.uni`, `rma.mh`, `rma.peto`, `rma.glmm`

Examples

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

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

### suppose that for a particular study, yi and vi are known (i.e., have
### already been calculated) but the 2x2 table counts are not known; with
### replace=FALSE, the yi and vi values for that study are not replaced
dat[1:12,10:11] <- NA
dat[13,4:7] <- NA
dat <- escalc(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat, replace=FALSE)
dat

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

---

fitstats *Fit Statistics and Information Criteria*

Description

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

Usage

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

Arguments

`object`	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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

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

See Also

fitstats.rma

---

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

Description

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

Usage

```
## S3 method for class 'rma'
fitstats(object, REML, ...)
## S3 method for class 'rma'
logLik(object, REML, ...)
## S3 method for class 'rma'
deviance(object, REML, ...)
## S3 method for class 'rma'
AIC(object, ..., k=2)
## S3 method for class 'rma'
BIC(object, ...)
```

Arguments

object	an object of class "rma".
REML	logical indicating whether the regular or restricted likelihood function should be used to obtain the fit statistics and information criteria. Defaults to the method of estimation used, that is TRUE if object was fitted with method = "REML" and FALSE otherwise.
k	numeric value specifying the penalty per parameter to be used. The default (k=2) is the classical AIC. See AIC for more details.
...	other arguments.

Value

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

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

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

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

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

Note

Note that variance components in the model (e.g., τ^2 in random/mixed-effects models) are counted as additional parameters in the calculation of the AIC and BIC. Also, the fixed effects are counted as parameters in the calculation of the AIC and BIC even when using REML estimation.

Author(s)

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

References

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

See Also

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

Examples

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

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

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

fitstats(res1)
fitstats(res2)

logLik(res1)
logLik(res2)
```

```
deviance(res1)
deviance(res2)

AIC(res1)
AIC(res2)

BIC(res1)
BIC(res2)
```

`fitted.rma` *Fitted Values for rma Objects*

Description

The function calculates the fitted values for objects of class `"rma"`.

Usage

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

Arguments

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

Value

A vector with the fitted values.

Note

The `predict.rma` function also provides standard errors and confidence intervals for the fitted values. Best linear unbiased predictions (BLUPs) that combine the fitted values based on the fixed effects and the estimated contributions of the random effects can be obtained with `blup.rma.uni` (only for objects of class `"rma.uni"`).

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

Author(s)

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

References

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

See Also

predict.rma, blup.rma.uni

Examples

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

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

---

forest *Forest Plots*

Description

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

Usage

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

Arguments

x	either an object of class "rma", a vector with the observed effect size or outcomes, or an object of class "cumul.rma". See 'Details'.
...	other arguments.

Details

Currently, methods exist for three types of situations.

In the first case, object x is a fitted model object coming from the rma.uni, rma.mh, or rma.peto functions. The corresponding method is then forest.rma.

Alternatively, object x can be a vector with observed effect size or outcomes. The corresponding method is then forest.default.

Finally, object x could be an object coming from the cumul.rma.uni, cumul.rma.mh, or cumul.rma.peto functions. The corresponding method is then forest.cumul.rma.

Author(s)

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

References

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

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

See Also

forest.rma, forest.default, forest.cumul.rma

---

forest.cumul.rma *Forest Plots for cumul.rma Objects*

Description

Function to create forest plots for objects of class "cumul.rma".

Usage

```
## S3 method for class 'cumul.rma'
forest(x, annotate=TRUE, xlim, alim, ylim, at, steps=5,
       level=x$level, digits=2, refline=0, xlab,
       ilab, ilab.xpos, ilab.pos,
       transf=FALSE, atransf=FALSE, targs, rows,
       efac=1, pch=15, psize=1,
       cex, cex.lab, cex.axis, ...)
```

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. If unspecified, the function tries to set the horizontal plot limits to some sensible values.
alim	the actual x axis limits. If unspecified, the function tries to set the x axis limits to some sensible values.
ylim	the y limits of the plot. If unspecified, the function tries to set the y axis limits to some sensible values.
at	position of the x axis tick marks and corresponding labels. If unspecified, the function tries to set the tick mark positions/labels to some sensible values.
steps	the number of tick marks for the x axis (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 specifying the number of decimal places to which the tick mark labels of the x axis and the annotations should be rounded (default is 2). Can also be a vector of two integers, the first specifying the number of decimal places for the annotations, the second for the x axis labels.
refline	value at which a vertical 'reference' line should be drawn (default is 0). The line can be suppressed by setting this argument to NA.
xlab	title for the x axis. If unspecified, the function tries to set an appropriate axis title.
ilab	optional vector or matrix of character strings providing additional information about the studies.
ilab.xpos	vector of numerical value(s) specifying the x axis position(s) of the character vector(s) given via ilab (must be specified if ilab is specified).
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 means left aligned). If unspecified, the default is to center the labels.
transf	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	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.
rows	optional vector specifying the rows (or more generally, the horizontal positions) for plotting the outcomes. If unspecified, the function sets this value automatically. Can also be a single value specifying the row (horizontal position) of the first outcome (the remaining outcomes are then plotted below this starting row).
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. Can also be a vector of values.
psize	optional vector with point sizes for the observed effect sizes or outcomes. Default is 1.
cex	optional character and symbol expansion factor. If unspecified, the function tries to set this to a sensible value.
cex.lab	optional expansion factor for the x axis titel. If unspecified, the function tries to set this to a sensible value.
cex.axis	optional expansion factor for the x axis labels. If unspecified, the function tries to set this to a sensible value.
...	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 (one should call the function again after adjusting the window size, so that the label/symbol sizes can be 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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

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

Lau, J., Schmid, C. H., & Chalmers, T. C. (1995). Cumulative meta-analysis of clinical trials builds evidence for exemplary medical care. *Journal of Clinical Epidemiology*, **48**, 45–57.

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

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

See Also

forest, cumul

Examples

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

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

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

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

```
forest(x, alim=c(-2,1))

### meta-analysis of the (log) relative risks using the Mantel-Haenszel method
res <- rma.mh(ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg,
              measure="RR", slab=paste(author, year, sep=", "))
x <- cumul(res, order=order(dat$year))
forest(x, alim=c(-2,1))
```

`forest.default` *Forest Plots*

Description

Function to create forest plots for a given set of data.

Usage

```
## Default S3 method:
forest(x, vi, sei, ci.lb, ci.ub, annotate=TRUE, showweight=FALSE,
       xlim, alim, ylim, at, steps=5,
       level=95, digits=2, refline=0, xlab,
       slab, ilab, ilab.xpos, ilab.pos,
       subset, transf=FALSE, atransf=FALSE, targs, rows,
       efac=1, pch=15, psize, cex, cex.lab, cex.axis, ...)
```

Arguments

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

`at`	position of the x axis tick marks and corresponding labels. If unspecified, the function tries to set the tick mark positions/labels to some sensible values.
`steps`	the number of tick marks for the x axis (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 specifying the number of decimal places to which the tick mark labels of the x axis and the annotations should be rounded (default is 2). Can also be a vector of two integers, the first specifying the number of decimal places for the annotations, the second for the x axis labels.
`refline`	value at which a vertical 'reference' line should be drawn (default is 0). The line can be suppressed by setting this argument to `NA`.
`xlab`	title for the x axis. If unspecified, the function tries to set an appropriate axis title.
`slab`	optional vector with unique labels for the k studies. If unspecified, simple labels are created within the function. To suppress labels, set this argument to `NA`.
`ilab`	optional vector 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` (must be specified if `ilab` is specified).
`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). If unspecified, the default is to center the labels.
`subset`	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`	optional argument specifying the name of a function that should be used to transform the observed effect sizes or outcomes and corresponding confidence interval bounds (e.g., `transf=exp`). Defaults to `FALSE`, which means that no transformation is used.
`atransf`	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`.
`rows`	optional vector specifying the rows (or more generally, the horizontal positions) for plotting the outcomes. If unspecified, the function sets this value automatically. Can also be a single value specifying the row (horizontal position) of the first outcome (the remaining outcomes are then plotted below this starting row).
`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. Can also be a vector of values.

psize	optional vector with point sizes for the observed effect sizes or outcomes. If unspecified, the point sizes are drawn proportional to the precision of the estimates.
cex	optional character and symbol expansion factor. If unspecified, the function tries to set this to a sensible value.
cex.lab	optional expansion factor for the x axis titel. If unspecified, the function tries to set this to a sensible value.
cex.axis	optional expansion factor for the x axis labels. If unspecified, the function tries to set this to a sensible value.
...	other arguments.

Details

The plot shows the individual observed effect sizes or outcomes with corresponding confidence intervals. To use the function, one should specify the observed effect sizes or outcomes (via the x argument) together with the corresponding sampling variances (via the vi argument) or with the corresponding standard errors (via the sei argument). Alternatively, one can specify the observed effect sizes or outcomes together with the corresponding confidence interval bounds (via the ci.lb and ci.ub arguments).

With the transf argument, the observed effect sizes or outcomes and corresponding confidence interval bounds can be transformed with an arbitrary function. For example, when plotting log odds ratios, then one could use transf=exp to obtain a forest plot showing the odds ratios. Alternatively, one can use the atransf argument to transform the x axis labels and annotations (e.g., atransf=exp). The examples below illustrate the use of these arguments.

By default, the studies are ordered from top to bottom (i.e., the first study in the dataset will be placed in row k, the second study in row $k-1$, and so on, until the last study, which is placed in the first row). The studies can be reordered with the subset argument (by specifying a vector with indices with the desired order).

Summary estimates can also be added to the plot with the addpoly function. See the documentation for that function for examples.

Note

The function tries to set some sensible values for the optional arguments, but it may be necessary to tweak these in certain circumstances. In particular, if the number of studies is very large, the labels, annotations, and symbols may become quite small and impossible to read. Stretching the plot window vertically may then provide a more readable figure (one should call the function again after adjusting the window size, so that the label/symbol sizes can be 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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

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

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

See Also

forest, forest.rma, addpoly

Examples

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

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

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

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

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

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

---

forest.rma *Forest Plots for rma Objects*

Description

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

Usage

```
## S3 method for class 'rma'
forest(x, annotate=TRUE, addfit=TRUE, addcred=FALSE, showweight=FALSE,
       xlim, alim, ylim, at, steps=5,
       level=x$level, digits=2, refline=0, xlab,
       slab, mlab, ilab, ilab.xpos, ilab.pos,
       order, transf=FALSE, atransf=FALSE, targs, rows,
       efac=1, pch=15, psize, col="darkgray", border="darkgray",
       cex, cex.lab, cex.axis, ...)
```

Arguments

x	an object of class "rma".
annotate	logical specifying whether annotations should be added to the plot (default is TRUE).
addfit	logical specifying whether the summary estimate (for models without moderators) or fitted values (for models with moderators) should be added to the plot. See 'Details'.
addcred	logical specifying whether the bounds of the credibility interval should be added to the plot. See 'Details'.
showweight	logical specifying whether the annotations should also include the weights given to the observed effects or outcomes during the model fitting (default is FALSE).
xlim	horizontal limits of the plot region. If unspecified, the function tries to set the horizontal plot limits to some sensible values.
alim	the actual x axis limits. If unspecified, the function tries to set the x axis limits to some sensible values.
ylim	the y limits of the plot. If unspecified, the function tries to set the y axis limits to some sensible values.
at	position of the x axis tick marks and corresponding labels. If unspecified, the function tries to set the tick mark positions/labels to some sensible values.
steps	the number of tick marks for the x axis (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 specifying the number of decimal places to which the tick mark labels of the x axis and the annotations should be rounded (default is 2). Can also be a vector of two integers, the first specifying the number of decimal places for the annotations, the second for the x axis labels.
refline	value at which a vertical 'reference' line should be drawn (default is 0). The line can be suppressed by setting this argument to NA.
xlab	title for the x axis. If unspecified, the function tries to set an appropriate axis title.
slab	optional vector with unique labels for the k studies. If unspecified, the labels are either taken from the object (if study labels were specified) or simple labels are created within the function. To suppress labels, set this argument to NA.
mlab	optional character string giving a label to the summary estimate from a fixed- or random-effects model. If unspecified, the label is created within the function.
ilab	optional vector or matrix of character strings providing additional information about the studies.
ilab.xpos	vector of numerical value(s) specifying the x axis position(s) of the character vector(s) given via ilab (must be specified if ilab is specified).
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). If unspecified, the default is to center the labels.

order	optional character string specifying how the studies should be ordered. See 'Details'.
transf	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	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.
rows	optional vector specifying the rows (or more generally, the horizontal positions) for plotting the outcomes. If unspecified, the function sets this value automatically. Can also be a single value specifying the row (horizontal position) of the first outcome (the remaining outcomes are then plotted below this starting row).
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. Can also be a vector of values.
psize	optional vector with point sizes for the observed effect sizes or outcomes. If unspecified, the point sizes are drawn proportional to the precision of the estimates.
cex	optional character and symbol expansion factor. If unspecified, the function tries to set this to a sensible value.
cex.lab	optional expansion factor for the x axis titel. If unspecified, the function tries to set this to a sensible value.
cex.axis	optional expansion factor for the x axis labels. If unspecified, the function tries to set this to a sensible value.
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 (i.e., for models without moderators), a polygon is added to the bottom of the forest plot, showing the summary estimate based on the model (with the outer edges of the polygon indicating the confidence interval limits). For random-effects models and if addcred=TRUE, a dotted line indicates the (approximate) bounds of the credibility interval (the interval indicates where level % of the true effects are expected to fall).

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

With the `transf` argument, the observed effect sizes or outcomes, summary estimate, fitted values, and confidence/credibility interval bounds can be transformed with an arbitrary function. For example, when plotting log odds ratios, one could use `transf=exp` to obtain a forest plot showing the odds ratios. Alternatively, one can use the `atransf` argument to transform the x axis labels and annotations (e.g., `atransf=exp`). The examples below illustrate the use of these arguments.

By default, the studies are ordered from top to bottom (i.e., the first study in the dataset will be placed in row k, the second study in row $k-1$, and so on, until the last study, which is placed in the first row). The studies can be reordered with the `order` argument:

- `order="obs"`: the studies are ordered by the observed effect sizes,
- `order="fit"`: the studies are ordered by the fitted values,
- `order="prec"`: the studies are ordered by their sampling variances,
- `order="resid"`: the studies are ordered by the size of their residuals,
- `order="rstandard"`: the studies are ordered by the size of their standardized residuals,
- `order="abs.resid"`: the studies are ordered by the size of their absolute residuals,
- `order="abs.rstandard"`: the studies are ordered by the size of their absolute standardized residuals.

Alternatively, it is also possible to set `order` equal to a vector with indices specifying the desired order (see examples below).

Additional summary estimates can also be added to the plot with the `addpoly` function. See the documentation for that function for examples.

Note

The function tries to set some sensible values for the optional arguments, but it may be necessary to tweak these in certain circumstances. In particular, if the number of studies is quite large, the labels, annotations, and symbols may become quite small and impossible to read. Stretching the plot window vertically may then provide a more readable figure (one should call the function again after adjusting the window size, so that the label/symbol sizes can be 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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

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

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

See Also

forest, forest.default, addpoly

Examples

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

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

### default forest plot of the log relative risks and summary estimate
forest(res)

### summary estimate in row -1; studies in rows 13 through 1; horizontal
### lines in rows 0 and k+1; and two extra lines of space at the top
text(x=-8.2, y=-1:16, -1:16, pos=4, cex=.5)

### several forest plots illustrating the use of various arguments
forest(res, cex=.8)
forest(res, cex=.8, addcred=TRUE)
forest(res, cex=.8, alim=c(-3,3))
forest(res, cex=.8, order="prec", alim=c(-3,3))
forest(res, cex=.8, order=order(dat.bcg$ablat), addcred=TRUE)

### adjust xlim values to see how that changes the plot
forest(res)
par("usr")[1:2] ### this shows what xlim values were chosen by default
forest(res, xlim=c(-16,14))
forest(res, xlim=c(-18,10))
forest(res, xlim=c(-10,10))

### illustrate transf and atransf arguments
forest(res, transf=exp, alim=c(0,6), xlim=c(-8,12), refline=1, cex=.8)
forest(res, atransf=exp, at=log(c(.05,.25,1,4,20)), xlim=c(-8,8),
       order="prec", showweight=TRUE, cex=.8)

### forest plot with extra annotations
forest(res, atransf=exp, at=log(c(.05, .25, 1, 4)), xlim=c(-16,6),
       ilab=cbind(dat.bcg$tpos, dat.bcg$tneg, dat.bcg$cpos, dat.bcg$cneg),
       ilab.xpos=c(-9.5,-8,-6,-4.5), cex=.75)
op <- par(cex=.75, font=2)
text(c(-9.5,-8,-6,-4.5), 15, c("TB+", "TB-", "TB+", "TB-"))
```

```
text(c(-8.75,-5.25),     16, c("Vaccinated", "Control"))
text(-16,                15, "Author(s) and Year",     pos=4)
text(6,                  15, "Relative Risk [95% CI]", pos=2)
par(op)

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

### forest plot with observed and fitted values
forest(res, xlim=c(-9,5), order="fit", cex=.8, ilab=dat.bcg$ablat,
       ilab.xpos=-4, atransf=exp, at=log(c(.05,.25,1,4)))
op <- par(font=2)
text(-9, 15, "Author(s) and Year",    pos=4, cex=.8)
text( 5, 15, "Observed RR [95% CI]",  pos=2, cex=.8)
text(-4, 15, "Latitude",                     cex=.8)
par(op)

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

### for more complicated plots, the ylim and rows arguments may be useful
forest(res)
forest(res, ylim=c(-1.5, 16)) ### the default
forest(res, ylim=c(-1.5, 20)) ### extra space in plot
forest(res, ylim=c(-1.5, 20), rows=c(17:15, 12:6,3:1)) ### set positions

### forest plot with subgrouping of studies
forest(res, xlim=c(-16, 6), at=log(c(.05, .25, 1, 4)), atransf=exp,
       ilab=cbind(dat.bcg$tpos, dat.bcg$tneg, dat.bcg$cpos, dat.bcg$cneg),
       ilab.xpos=c(-9.5,-8,-6,-4.5), cex=.75, ylim=c(-1, 21),
       order=order(dat.bcg$alloc), rows=c(1:2,5:11,14:17))
op <- par(cex=.75, font=4)
text(-16, c(18,12,3), c("Systematic Allocation", "Random Allocation",
                        "Alternate Allocation"), pos=4)
par(cex=.75, font=2)
text(c(-9.5,-8,-6,-4.5), 20, c("TB+", "TB-", "TB+", "TB-"))
text(c(-8.75,-5.25),     21, c("Vaccinated", "Control"))
text(-16,                20, "Author(s) and Year",     pos=4)
text(6,                  20, "Relative Risk [95% CI]", pos=2)
par(op)

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

`fsn` *Fail-Safe N (File Drawer Analysis)*

Description

Function to calculate the fail-safe N.

Usage

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

Arguments

yi	vector with the observed effect sizes or outcomes.
vi	vector with the corresponding sampling variances.
sei	vector with the corresponding standard errors. (note: only one of the two, vi or sei, needs to be specified)
data	optional data frame containing the variables given to the arguments above.
type	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 undefined, then the target average effect size will be equal to the observed average effect size divided by 2.
subset	optional vector indicating the subset of studies that should be used for the calculation. This can be a logical vector of length k or a numeric vector indicating the indices of the observations to include.
digits	integer specifying the number of decimal places to which the printed results should be rounded (default is 4).

Details

The Rosenthal method (sometimes called a 'file drawer analysis') calculates the number of studies averaging null results that would have to be added to the given set of observed outcomes to reduce the combined significance level (p-value) to a 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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

Rosenthal, R. (1979). The "file drawer problem" and tolerance for null results. *Psychological Bulletin*, **86**, 638–641.

Orwin, R. G. (1983). A fail-safe N for effect size in meta-analysis. *Journal of Educational Statistics*, **8**, 157–159.

Rosenberg, M. S. (2005). The file-drawer problem revisited: A general weighted method for calculating fail-safe numbers in meta-analysis. *Evolution*, **59**, 464–468.

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

See Also

ranktest, trimfill

Examples

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

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

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

`funnel` *Funnel Plots*

Description

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

Usage

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

Arguments

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

Details

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

Author(s)

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

References

Light, R. J., & Pillemer, D. B. (1984). *Summing up: The science of reviewing research*. Cambridge, MA: Harvard University Press.

Sterne, J. A. C., & Egger, M. (2001). Funnel plots for detecting bias in meta-analysis: Guidelines on choice of axis. *Journal of Clinical Epidemiology*, **54**, 1046–1055.

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

See Also

`funnel.rma`

funnel.rma *Funnel Plots for rma Objects*

Description

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

Usage

```
## S3 method for class 'rma'
funnel(x, yaxis="sei", xlim, ylim, xlab, ylab,
       steps=5, at, atransf=FALSE, targs, digits,
       level=x$level, addtau2=FALSE, type="rstandard",
       back="lightgray", shade="white", hlines="white",
       refline, pch=19, pch.fill=21, ci.res=1000, ...)
```

Arguments

x	an object of class "rma".
yaxis	either "sei", "vi", "seinv", "vinv", "ni", "ninv", "sqrtni", "sqrtninv", or "lni" to indicate what values should be placed on the y axis. See 'Details'.
xlim	x axis limits. If unspecified, the function tries to set the x axis limits to some sensible values.
ylim	y axis limits. If unspecified, the function tries to set the y axis limits to some sensible values.
xlab	title for the x axis. If unspecified, the function tries to set an appropriate axis title.
ylab	title for the y axis. If unspecified, the function tries to set an appropriate axis title.
steps	the number of tick marks for the y axis (default is 5).
at	position of the x axis tick marks and corresponding labels. If unspecified, the function tries to set the tick mark positions/labels to some sensible values.
atransf	optional argument specifying the name of a function that should be used to transform the x 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 atransf.
digits	integer specifying the number of decimal places to which the tick mark labels of the x and y axis should be rounded. Can also be a vector of two integers, the first specifying the number of decimal places for the x axis, the second for the y axis labels (e.g., c(2,3)). If unspecified, the function tries to set the argument to some sensible values.
level	numerical value between 0 and 100 specifying the level of the pseudo confidence interval region (the default is to take the value from the object). May also be a vector of values to obtain multiple regions. See 'Examples'.

addtau2	logical to indicate whether the amount of heterogeneity should be accounted for when drawing the pseudo confidence interval region (default is FALSE). Ignored when the model includes moderators and residuals are plotted.
type	either "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 to draw the vertical reference line and, if drawn, where the pseudo confidence interval should be centered. If unspecified, the reference line is drawn at the fixed- or random-effects model estimate when the model does not include moderators and at zero when moderators are included (and therefore residuals are plotted).
pch	plotting symbol to use for the observed effect sizes or outcomes. By default, a solid circle is used. Can also be a vector of values. See points for other options.
pch.fill	plotting symbol to use for the effect sizes or outcomes filled in by the trim and fill method. By default, a circle is used. Only relevant when plotting an object created by the trimfill function.
ci.res	integer specifying the number of y axis values at which to calculate the bounds of the pseudo confidence interval. The default is 1000, which usually provides a sufficient resolution for the plotting.
...	other arguments.

Details

For fixed- and random-effects models (i.e., models not involving moderators), the plot shows the individual observed effect sizes or outcomes on the x axis against the corresponding standard errors (i.e., the square root of the sampling variances) on the y axis. A vertical line indicates the estimate based on the model. A pseudo confidence interval region is drawn around this value with bounds equal to $\pm 1.96SE$, where SE is the standard error value from the y axis. If addtau2=TRUE, then the bounds of the pseudo confidence interval region are equal to $\pm 1.96\sqrt{SE^2 + \tau^2}$, where τ^2 is the amount of heterogeneity as estimated by the model.

For models involving moderators, the plot shows the residuals on the x axis against their corresponding standard errors. Either the usual or deleted residuals can be used for that purpose (set via the type argument). See residuals.rma for more details on the different types of residuals.

Instead of placing the standard error value on the y axis, several other options are available by setting the yaxis argument to:

- yaxis="vi" for the sampling variance,
- yaxis="seinv" for the inverse of the standard error,
- yaxis="vinv" for the inverse of the sampling variance,
- yaxis="ni" for the sample size,
- yaxis="ninv" for the inverse of the sample size,

- `yaxis="sqrtni"` for the square-root sample size,
- `yaxis="sqrtninv"` for the inverse of the square-root sample size,
- `yaxis="lni"` for the log of the sample size.

However, only when `yaxis="sei"` (the default) will the pseudo confidence region have the expected (upside-down) funnel shape with straight lines. Also, when placing (a function of) the sample size on the y axis, then the pseudo confidence region cannot be drawn. See Sterne and Egger (2001) for more details on the choice of the y axis.

If the object passed to the function comes from the `trimfill` function, the effect sizes or outcomes that are filled in by the trim and fill method are also added to the funnel plot. The symbol to use for plotting the filled in values can then be specified via the `pch.fill` argument.

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

Note

Placing (a function of) the sample size on the y axis (i.e., using `yaxis="ni"`, `yaxis="ninv"`, `yaxis="sqrtni"`, `yaxis="sqrtninv"`, or `yaxis="lni"`) is only possible when information about the sample sizes is actually stored within the object passed to the `funnel` function. That should automatically be the case when the observed outcomes were computed with the `escalc` function or when the observed outcomes were computed within the model fitting function. The only time this will not automatically be the case is when `rma.uni` was used together with the `yi` and `vi` arguments and the `yi` and `vi` values were *not* computed with `escalc`. In that case, it is still possible to pass information about the sample sizes to the `rma.uni` function (i.e., use `rma.uni(yi, vi, ni)`).

Author(s)

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

References

Light, R. J., & Pillemer, D. B. (1984). *Summing up: The science of reviewing research*. Cambridge, MA: Harvard University Press.

Peters, J. L., Sutton, A. J., Jones, D. R., Abrams, K. R., & Rushton, L. (2008). Contour-enhanced meta-analysis funnel plots help distinguish publication bias from other causes of asymmetry. *Journal of Clinical Epidemiology*, **61**, 991–996.

Sterne, J. A. C., & Egger, M. (2001). Funnel plots for detecting bias in meta-analysis: Guidelines on choice of axis. *Journal of Clinical Epidemiology*, **54**, 1046–1055.

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

See Also

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

Examples

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

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

### standard funnel plot
funnel(res)

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

### contour-enhanced funnel plot centered at 0 (see Peters et al., 2008)
funnel(res, level=c(90, 95, 99), shade=c("white", "gray", "darkgray"),
       refline=0, atransf=exp, at=log(c(.10, .25, .5, 1, 2, 4, 10)))

### 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 plot of the residuals
funnel(res)

### simulate a large meta-analytic dataset (correlations with rho = 0.2)
### with no heterogeneity or publication bias; then try out some various
### versions of the funnel plot

gencor <- function(rhoi, ni) {
   x1 <- rnorm(ni, mean=0, sd=1)
   x2 <- rnorm(ni, mean=0, sd=1)
   x3 <- rhoi*x1 + sqrt(1-rhoi^2)*x2
   cor(x1, x3)
}

k  <- 200                               ### number of studies to simulate
ni <- round(rchisq(k, df=2) * 20 + 20)  ### simulate sample sizes (skewed distribution)
ri <- mapply(gencor, rep(0.2,k), ni)    ### simulate correlations

res <- rma(ri=ri, ni=ni, measure="ZCOR", method="FE")

funnel(res, yaxis="sei")
funnel(res, yaxis="vi")
funnel(res, yaxis="seinv")
funnel(res, yaxis="vinv")
funnel(res, yaxis="ni")
funnel(res, yaxis="ninv")
funnel(res, yaxis="sqrtni")
funnel(res, yaxis="sqrtninv")
funnel(res, yaxis="lni")
```

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

Description

The functions calculate 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$digits, ...)
## S3 method for class 'infl.rma.uni'
print(x, digits=x$digits, ...)
## S3 method for class 'rma.uni'
cooks.distance(model, ...)
## S3 method for class 'rma.uni'
dfbetas(model, ...)
## S3 method for class 'rma.uni'
hatvalues(model, ...)
```

Arguments

`model`	an object of class `"rma.uni"`.
`x`	an object of class `"infl.rma.uni"` (for `print`).
`digits`	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 `influence` function calculates the following leave-one-out diagnostics for each study:

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

The diagonal elements of the hat matrix and the weights (in %) given to the observed effects or outcomes during the model fitting are also provided (except for their scaling, the hat values and weights are the same for models without moderators, but will differ when moderators are included).

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

The DFFITS value essentially indicates how many standard deviations the predicted (average) effect for the i^{th} study changes after excluding the i^{th} study from the model fitting.

Cook's distance can be interpreted as the Mahalanobis distance between the entire set of predicted values once with the i^{th} study included and once with the i^{th} study excluded from the model fitting.

The covariance ratio is defined as the determinant of the variance-covariance matrix of the parameter estimates based on the dataset with the i^{th} study removed divided by the determinant of the variance-covariance matrix of the parameter estimates based on the complete dataset. A value below 1 therefore indicates that removal of the i^{th} study yields more precise estimates of the model coefficients.

The leave-one-out amount of (residual) heterogeneity is the estimated value of τ^2 based on the dataset with the i^{th} study removed. Note that this is always equal to `0` for fixed-effects models.

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

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

A study may be considered to be 'influential' if at least one of the following is true:

- The absolute DFFITS value is larger than $3\sqrt{p/(k-p)}$, where p is the number of model coefficients and k the number of studies.
- The lower tail area of a chi-square distribution with p degrees of freedom cut off by the Cook's distance is larger than 50%.
- The hat value is larger than $3(p/k)$.
- Any DFBETAS value is larger than 1.

Studies which are considered influential with respect to any of these measures are marked with an asterisk. Note that the chosen cut-offs are (somewhat) arbitrary. Substantively informed judgment should always be used when examining the influence of each study on the results.

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.
`is.infl`	a logical vector indicating whether a particular study is considered to be 'influential'.
`...`	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 function).

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

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

Author(s)

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

References

Belsley, D. A., Kuh, E., & Welsch, R. E. (1980). *Regression diagnostics*. New York: Wiley.

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

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

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

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

See Also

plot.infl.rma.uni, rstudent.rma.uni, weights.rma.uni

Examples

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

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

cooks.distance(res)
dfbetas(res)
```

```
hatvalues(res)
```

---

labbe *L'Abbe Plots*

Description

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

Usage

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

Arguments

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

Details

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

Author(s)

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

References

L'Abbe, K. A., Detsky, A. S., & O'Rourke, K. (1987). Meta-analysis in clinical research. *Annals of Internal Medicine*, **107**, 224–233.

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

See Also

labbe.rma

labbe.rma *L'Abbe Plots for rma Objects*

Description

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

Usage

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

Arguments

x	an object of class "rma". See 'Details'.
xlim	x axis limits. If unspecified, the function tries to set the x axis limits to some sensible values.
ylim	y axis limits. If unspecified, the function tries to set the y axis limits to some sensible values.
xlab	title for the x axis. If unspecified, the function tries to set an appropriate axis title.
ylab	title for the y axis. If unspecified, the function tries to set an appropriate axis title.
add	See below and the documentation of the escalc function for more details.
to	See below and the documentation of the escalc function for more details.
transf	optional argument specifying the name of a function that should be used to transform the outcomes (e.g., transf=exp). Defaults to FALSE, which means that no transformation is used.
targs	optional arguments needed by the function specified under transf.
pch	plotting symbol to use for the outcomes. By default, a filled circle is used. Can also be a vector of values. See points for other options.
psize	optional vector with point sizes for the outcomes. If unspecified, the point sizes are drawn proportional to the precision of the estimates.
bg	color to use for filling the plotting symbol (default is "gray"). Can also be a vector of values. Set to NA to make the plotting symbols transparent.
...	other arguments.

Details

The model specified by x must be a model without moderators (i.e., either a fixed- or a random-effects model) fitted with either the rma.uni, rma.mh, rma.peto, or rma.glmm functions. Moreover, the model must be fitted with measure set equal to "RD" (for risk differences), "RR" (for relative risks), "OR" (for odds ratios), "AS" (for arcsine transformed risk differences), "IRR" (for incidence rate ratios), "IRD" (for incidence rate differences), or "IRSD" (for square-root transformed incidence rate differences).

The function calculates the arm-level outcomes for the two experimental groups (e.g., treatment and control groups) and plots them against each other. In particular, the function plots the raw proportions of the two groups again each other when analyzing risk differences, the log of the proportions when analyzing (log) relative risks, the log odds when analyzing (log) odds ratios, the arcsine transformed proportions when analyzing arcsine transformed risk differences, the raw incidence rates when analyzing incidence rate differences, the log of the incidence rates when analyzing (log) incidence rate ratios, and the square-root transformed incidence rates when analyzing square-root transformed incidence rate differences. The transf argument can be used to transform these values (for example, transf=exp to transform the log of the proportions back to raw proportions).

As described under the documentation for the escalc function, zero cells can lead to problems when calculating particular outcomes. Adding a small constant to the cells of the 2x2 tables is a common solution to this problem. By default, the functions adopts the same method for handling zero cells as was done when fitting the model.

The size of the points is drawn proportional to the precision (inverse standard error) of the outcomes. The solid line corresponds to identical outcomes in the two groups (i.e., the absence of a difference between the two groups). The dashed line indicates the estimated effect based on the fitted model.

Author(s)

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

References

L'Abbe, K. A., Detsky, A. S., & O'Rourke, K. (1987). Meta-analysis in clinical research. *Annals of Internal Medicine*, **107**, 224–233.

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

See Also

rma.uni, rma.mh, rma.peto, rma.glmm

Examples

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

### meta-analysis of the log relative risks using a random-effects model
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg,
```

```
            data=dat.bcg, measure="RR", method="REML")

### default plot
labbe(res)

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

`leave1out` *Leave-One-Out Diagnostics*

Description

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

Usage

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

Arguments

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

Details

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

Author(s)

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

References

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

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

See Also

`leave1out.rma.uni`, `leave1out.rma.mh`, `leave1out.rma.peto`

leave1out.rma.mh *Leave-One-Out Diagnostics for rma.mh and rma.peto Objects*

Description

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

Usage

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

Arguments

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

Value

An object of class `"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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

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

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

See Also

leave1out

Examples

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

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

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

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

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

---

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

Description

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

Usage

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

Arguments

`x`	an object of class `"rma.uni"`.
`digits`	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`	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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

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

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

See Also

leave1out, influence.rma.uni

Examples

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

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

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

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

---

llplot *Plot Log Likelihood of a Parameter Corresponding to an Outcome or Effect Size Measure*

Description

Function to plot the log likelihood of a certain parameter corresponding to an outcome or effect size measure given the study data.

Usage

```
llplot(measure="OR", ai, bi, ci, di, n1i, n2i, data, subset, drop00=TRUE,
       xvals=1000, xlim, ylim, xlab, ylab, scale=TRUE,
       lty, lwd, col, level=99.99, refline=0, ...)
```

Arguments

measure	a character string indicating for which effect size or outcome measure the log likelihood should be calculated. See 'Details' for possible options and how the data should be specified.
ai	vector to specify the 2x2 table frequencies (upper left cell).
bi	vector to specify the 2x2 table frequencies (upper right cell).
ci	vector to specify the 2x2 table frequencies (lower left cell).
di	vector to specify the 2x2 table frequencies (lower right cell).
n1i	vector to specify the group sizes or row totals (first group/row).
n2i	vector to specify the group sizes or row totals (second group/row).
data	optional data frame containing the variables given to the arguments above.

subset	optional vector indicating the subset of studies that should be used. This can be a logical vector or a numeric vector indicating the indices of the studies to include.
drop00	logical indicating whether studies with no cases (or only cases) in both groups should be dropped. See 'Details'.
xvals	integer specifying for how many distinct values of the (log) odds ratio the log likelihood should be evaluated.
xlim	x axis limits. If unspecified, the function tries to set the x axis limits to some sensible values.
ylim	y axis limits. If unspecified, the function tries to set the y axis limits to some sensible values.
xlab	title for the x axis. If unspecified, the function tries to set an appropriate axis title.
ylab	title for the y axis. If unspecified, the function tries to set an appropriate axis title.
scale	logical indicating whether the log likelihood values should be scaled, so that the total area under each curve is approximately equal to 1.
lty	the line type (either a single value or a vector of length k). If unspecified, the function uses solid lines for tables where the MLE of the odds ratio is finite and dashed/dotted lines otherwise.
lwd	the line width (either a single value or a vector of length k). If unspecified, the function sets the widths according to the sampling variances (so that the line is thicker for more precise studies and vice-versa).
col	the line color (either a single value or a vector of length k). If unspecified, the function uses various gray shades according to the sampling variances (so that darher shades are used for more precise studies and vice-versa).
level	numerical value between 0 and 100 specifying the plotting limits for each density line in terms of the confidence interval (default is 99.99).
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.
...	other arguments.

Details

At the moment, the function only accepts measure="OR". For each 2x2 table, the function then plots the log likelihood of the (log) odds ratio based on the non-central hypergeometric distribution. Since studies with no cases (or only cases) in both groups have a flat likelihood and are not informative about the odds ratio, they are dropped by default (i.e., drop00=TRUE). For studies that have a single zero count, the MLE of the odds ratio is infinite and these likelihoods are indicated by a dashed line.

Author(s)

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

References

van Houwelingen, H. C., Zwinderman, K. H., & Stijnen, T. (1993). A bivariate approach to meta-analysis. *Statistics in Medicine*, **12**, 2273–2284.

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

See Also

rma.glmm

Examples

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

### create plot
llplot(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg)

### load histamine H2 antagonist data
data(dat.collins1985a)

### create plot (Figure 2 in van Houwelingen, Zwinderman, & Stijnen, 1993)
llplot(measure="OR", ai=xci, n1i=nci, ci=xti, n2i=nti, data=dat.collins1985a,
       lwd=1, refline=NA, xlim=c(-4,4), drop00=FALSE)
```

---

metafor.news *Read News File of the Metafor Package*

Description

Read news file of the **metafor-package**.

Usage

```
metafor.news()
```

Details

The function is just a wrapper for `news(package="metafor")` which parses and displays the 'NEWS' file of the package.

Author(s)

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

References

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

Examples

```
metafor.news()
```

`permutest` *Permutation Tests*

Description

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

Usage

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

Arguments

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

Details

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

Author(s)

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

References

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

See Also

`permutest.rma.uni`

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

Description

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

Usage

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

Arguments

`x`	an object of class `"rma.uni"`.
`exact`	logical indicating whether an exact permutation test should be carried out or not (default is `FALSE`). See 'Details'.
`iter`	integer specifying the number of iterations for the permutation test when not doing an exact test (default is 1000 iterations).
`progbar`	logical indicating whether a progress bar should be shown (default is `TRUE`).
`retpermdist`	logical indicating whether the permutation distributions of the test statistics should be returned (default is `FALSE`).
`digits`	integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
`...`	other arguments.

Details

For models without moderators, the permutation test is carried out by permuting the signs of the observed effect sizes or outcomes. The (two-sided) p-value of the permutation test is then equal to two times the proportion of times that the test statistic under the permuted data is as extreme or more extreme than under the actually observed data. See Follmann and Proschan (1999) for more details.

For models with moderators, the permutation test is carried out by permuting the rows of the design matrix. The (two-sided) p-value for a particular model coefficient is then equal to two times the proportion of times that the test statistic for the coefficient under the permuted data is as extreme or more extreme than under the actually observed data. Similarly, for the omnibus test, the p-value is the proportion of times that the test statistic for the omnibus test is as extreme or more extreme than the actually observed one. See Higgins and Thompson (2004) for more details.

If `exact=TRUE`, the function will try to carry out an exact permutation test. An exact permutation test requires fitting the model to each possible permutation once. However, the number of possible permutations increases rapidly with the number of outcomes/studies (i.e., k). For models without moderators, there are 2^k possible permutations of the signs. Therefore, for $k = 5$, there are 32

possible permutations, for $k = 10$, there are already 1024, and for $k = 20$, there are over one million permutations of the signs.

For models with moderators, the increase in the number of possible permutations may be even more severe. The total number of possible permutations of the design matrix is $k!$. Therefore, for $k = 5$, there are 120 possible permutations, for $k = 10$, there are 3,628,800, and for $k = 20$, there are over 10^{18} permutations of the design matrix.

Therefore, going through all possible permutations may become infeasible. Instead of using an exact permutation test, one can set `exact=FALSE` (which is also the default). In that case, the function approximates the exact permutation-based p-value(s) by going through a smaller number (as specified by the `iter` argument) of *random* permutations. Therefore, running the function twice on the same data will yield (slightly) different p-values. Setting `iter` sufficiently large ensures that the results become stable. Note that if `exact=FALSE` and `iter` is actually larger than the number of iterations required for an exact permutation test, then an exact test will be carried out.

For models with moderators, the exact permutation test actually only requires fitting the model to each *unique* permutation of the design matrix. The number of unique permutations will be (often much) smaller than $k!$ when the design matrix contains recurring rows. This may be the case when only including categorical moderators (i.e., factors) in the model or when any quantitative moderators included in the model only take on a small number of unique values. When `exact=TRUE`, the function therefore uses an algorithm to restrict the test to only the unique permutations of the design matrix, which may make the use of the exact test feasible even when k is large.

Value

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

`pval`	p-value(s) based on the permutation test.
`QMp`	p-value for the omnibus test of coefficients based on the permutation test.
`zval.perm`	values of the test statistics of the coefficients under the various permutations (only when `retpermdist=TRUE`).
`QM.perm`	values of the test statistic for the omnibus test of coefficients under the various permutations (only when `retpermdist=TRUE`).
`...`	some additional elements/values are passed on.

The results are formated and printed with the `print.permutest.rma.uni` function. One can also use `coef.permutest.rma.uni` to obtain the table with the model coefficients, corresponding standard errors, test statistics, p-values, and confidence interval bounds.

Note

It is important to note that the p-values obtained with permutation tests cannot reach conventional levels of statistical significance (i.e., $p \leq .05$) when k is very small. In particular, for models without moderators, the smallest possible (two-sided) p-value is .0625 when $k = 5$ and .03125 when $k = 6$. Therefore, k must be at least equal to 6 to reject the null hypothesis at $\alpha = .05$. For models with moderators, the smallest possible (two-sided) p-value for a particular model coefficient is .08333 when $k = 4$ and .01667 when $k = 5$ (assuming that each row in the design matrix is unique). Therefore, k must be at least equal to 5 to reject the null hypothesis at $\alpha = .05$.

When the number of permutations required for the exact test is so large as to be essentially indistinguishable from infinity (e.g., `factorial(200)`), the function will terminate with an error.

Author(s)

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

References

Follmann, D. A., & Proschan, M. A. (1999). Valid inference in random effects meta-analysis. *Biometrics*, **55**, 732–737.

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

Higgins, J. P. T., & Thompson, S. G. (2004). Controlling the risk of spurious findings from meta-regression. *Statistics in Medicine*, **23**, 1663–1682.

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

See Also

rma.uni, print.permutest.rma.uni

Examples

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

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

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

### permutation test (approximate and exact)
## Not run:

permutest(res)
permutest(res, exact=TRUE)

## End(Not run)

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

### permutation test (approximate only; exact not feasible)
## Not run:

permres <- permutest(res, iter=10000, retpermdist=TRUE)
permres

### histogram of permutation distribution for absolute latitude
### dashed horizontal line: the observed value of the test statistic
```

```
### red curve:  standard normal density
### blue curve: kernel density estimate of the permutation distribution
### note that the tail area under the permutation distribution is larger
### than under a standard normal density (hence, the larger p-value)
hist(permres$zval.perm[,2], breaks=120, freq=FALSE, xlim=c(-5,5), ylim=c(0,.4),
     main="Permutation Distribution", xlab="Value of Test Statistic")
abline(v=res$zval[2], lwd=2, lty="dashed")
abline(v=0, lwd=2)
curve(dnorm, from=-5, to=5, add=TRUE, lwd=2, col=rgb(1,0,0,alpha=.7))
lines(density(permres$zval.perm[,2]), lwd=2, col=rgb(0,0,1,alpha=.7))

## End(Not run)
```

`plot.infl.rma.uni` *Plot Method for infl.rma.uni Objects*

Description

Plot method for objects of class `"infl.rma.uni"`.

Usage

```
## S3 method for class 'infl.rma.uni'
plot(x, plotdfb=FALSE, dfbnew=FALSE, logcov=TRUE,
     which=1:8, layout, slab.style=1, las=0, pch=21, bg="black",
     bg.infl="red", col.na="lightgray", ...)
```

Arguments

`x`	an object of class `"infl.rma.uni"`.
`plotdfb`	logical indicating whether the DFBETAS values should be plotted (default is `FALSE`). Can also be a vector of integers to indicate for which coefficient(s) to plot the DFBETAS values.
`dfbnew`	logical indicating whether a new device should be opened for plotting the DFBETAS values (default is `FALSE`).
`logcov`	logical indicating whether the covariance ratios should be plotted on a log scale (default is `TRUE`).
`which`	vector of up to 8 integers indicating which plots to draw (see 'Details' for the numbers corresponding to the various plots).
`layout`	optional vector of two numbers, indicating the number of rows and columns for the layout of the figure.
`slab.style`	integer to indicate the style of the x axis labels: 1 = study number, 2 = study label, 3 = abbreviated study label. Note that study labels, even when abbreviated, may be too long to fit in the margin.)
`las`	integer between 0 and 3 to specify the alignment of the axis labels (see `par`). The most useful alternative to 0 is 3, so that the x axis labels are drawn vertical to the axis.

pch	plotting symbol to use. By default, a filled circle is used. See points for other options.
bg	color to use for filling the plotting symbol (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 NA values in between (default is "lightgray").
...	other arguments.

Details

The function plots the (1) externally standardized residuals, (2) DFFITS values, (3) Cook's distances, (4) covariance ratios, (5) leave-one-out τ^2 estimates, (6) leave-one-out (residual) heterogeneity test statistics, (7) hat values, and (8) weights. If 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).

A study may be considered to be 'influential' if at least one of the following is true:

- The absolute DFFITS value is larger than $3\sqrt{p/(k-p)}$, where p is the number of model coefficients and k the number of studies.
- The lower tail area of a chi-square distribution with p degrees of freedom cut off by the Cook's distance is larger than 50%.
- The hat value is larger than $3(p/k)$.
- Any DFBETAS value is larger than 1.

Studies which are considered influential with respect to any of these measures are indicated by the color specified for the bg.infl argument (the default is "red").

The cut-offs described above are indicated in the plot with horizontal reference lines. In addition, on the plot of the externally standardized residuals, horizontal reference lines are drawn at -1.96, 0, and 1.96. On the plot of the hat values, a horizontal reference line is drawn at p/k. Since the sum of the hat values is equal to p, the value p/k indicates equal hat values for all k studies. Finally, on the plot of weights, a horizontal reference line is drawn at $100/k$, corresponding to the value for equal weights (in %) for all k studies. Note that all weights will automatically be equal to each other when using unweighted model fitting. Also, the hat values will be equal to the weights values (except for their scaling) in models without moderators.

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

Author(s)

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

References

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

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

See Also

influence.rma.uni

Examples

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

### meta-analysis of the log relative risks using a mixed-effects model
### with two moderators (absolute latitude and publication year)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods = ~ ablat + year,
           data=dat.bcg, measure="RR", method="REML")
plot(influence(res))
plot(influence(res), which=1:4)
plot(influence(res), which=1:4, layout=c(4,1))
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, ...)
## S3 method for class 'rma.glmm'
plot(x, qqplot=FALSE, ...)
```

Arguments

x	an object of class "rma.uni", "rma.mh", or "rma.peto". The method is not yet implemented for objects of class "rma.glmm".
qqplot	logical indicating whether a normal QQ plot should be drawn (default is FALSE).
...	other arguments.

Details

Four plots are produced. If the model does not contain any moderators, then a forest plot, funnel plot, radial plot, and a plot of the standardized residuals is provided. If `qqplot=TRUE`, the last plot is replaced by a normal QQ plot of the standardized residuals.

If the model contains moderators, then a forest plot, funnel plot, plot of the standardized residuals against the fitted values, and a plot of the standardized residuals is provided. If `qqplot=TRUE`, the last plot is replaced by a normal QQ plot of the standardized residuals.

Note

If the number of studies is large, the forest plot may become quite impossible to read.

Author(s)

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

References

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

See Also

forest, funnel, radial, qqnorm.rma.uni

Examples

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

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

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

predict.rma *Predicted Values for rma Objects*

Description

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

Usage

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

Arguments

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

Details

For the fixed-effects model, predict(object) returns the estimated (average) outcome in the set of studies included in the meta-analysis. This is the same as the estimated intercept in the fixed-effects model.

For the random-effects model, predict(object) returns the estimated (average) outcome in the hypothetical population of studies from which the set of studies included in the meta-analysis are assumed to be a random selection. This is the same as the estimated intercept in the random-effects model.

For models including one or more moderators, predict(object) returns the estimated (average) outcomes for values of the moderator(s) equal to those of the k studies included in the meta-analysis (i.e., the fitted values for the k studies).

For models including q moderator variables, new moderator values for k_new new studies can be specified by setting `newmods` equal to an $k_new \times q$ matrix with the corresponding new moderator values. If the original model includes an intercept, then it does not need to be explicitly specified under `newmods`. Also, any factors in the original model get turned into the appropriate contrast variables within the `rma` function, so that `newmods` should actually include the values for the contrast variables. Examples are shown below.

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

Value

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

`pred`	predicted value(s).
`se`	corresponding standard error(s).
`ci.lb`	lower bound of the confidence interval(s).
`ci.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).
`X`	the moderator value(s) used to calculate the predicted values (only when `addx=TRUE`).
`...`	some additional elements/values.

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

Note

The predicted values are based only on the fixed effects of the model. Best linear unbiased predictions (BLUPs) that combine the fitted values based on the fixed effects and the estimated contributions of the random effects can be obtained with `blup.rma.uni` (only for objects of class `"rma.uni"`).

When using the `transf` option, the transformation is applied to the predicted values and the corresponding interval bounds. The standard errors are set equal to `NA`.

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

Author(s)

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

References

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

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

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

See Also

fitted.rma, blup.rma.uni

Examples

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

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

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

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

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

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

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

### predicted average relative risks for 10 and 60 degrees latitude in 1950 and 1980
predict(res, newmods=cbind(c(10,60,10,60),c(1950,1950,1980,1980)), transf=exp, addx=TRUE)

### mixed-effects model with two moderators (one of which is a factor)
res <- rma(ai=tpos, bi=tneg, ci=cpos, di=cneg, mods= ~ ablat + factor(alloc),
           measure="RR", data=dat.bcg, method="REML")

### examine how the factor was actually coded for the studies in the dataset
predict(res, addx=TRUE)

### predictd average relative risks at 30 degrees for the three factor levels
### note: the contrast (dummy) variables need to specified explicitly here
```

```
predict(res, newmods=c(30, 0, 0), addx=TRUE)   # for alternate  allocation
predict(res, newmods=c(30, 1, 0), addx=TRUE)   # for random     allocation
predict(res, newmods=c(30, 0, 1), addx=TRUE)   # for systematic allocation
```

`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`	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. This can be regarded as a pseudo-R^2 statistic (Raudenbush, 2009).

Value

The function does not return an object.

Author(s)

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

References

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

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

See Also

anova.rma.uni

`print.confint.rma` *Print Method for confint.rma Objects*

Description

Print method for objects of class `"confint.rma"`.

Usage

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

Arguments

`x`	an object of class `"confint.rma"`.
`digits`	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:

- estimate of the model coefficient or variance component
- lower bound of the confidence interval
- upper bound of the confidence interval

Value

The function does not return an object.

Author(s)

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

References

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

See Also

confint.rma.uni

`print.escalc` *Print and Summary Methods for escalc Objects*

Description

Print and summary methods for objects of class "escalc".

Usage

```
## S3 method for class 'escalc'
print(x, digits, ...)
## S3 method for class 'escalc'
summary(object, out.names=c("sei","zi","ci.lb","ci.ub"), var.names,
        append=TRUE, replace=TRUE, level=95, digits, transf=FALSE, ...)
## S3 method for class 'summary.escalc'
print(x, digits, ...)
```

Arguments

x	an object of class "escalc" or "summary.escalc".
object	an object of class "escalc".
digits	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 if possible).
out.names	character string with four elements, specifying the variable names for the standard errors, test statistics, and lower/upper confidence interval bounds.
var.names	character string with two elements, specifying the variable names for the observed outcomes and the sampling variances (the default is to take the value from the object if possible).
append	logical indicating whether the data frame specified via the object argument should be returned together with the additional variables that are calculated by the summary function (default is TRUE).
replace	logical indicating whether existing values for sei, zi, ci.lb, and ci.ub in the data frame should be replaced or not. Only relevant when the data frame already contains these variables. If replace=TRUE (the default), all of the existing values will be overwritten. If replace=FALSE, only NA values will be replaced.
level	numerical value between 0 and 100 specifying the confidence interval level (the default is 95).

transf	optional argument specifying the name of a function that should be used to transform the observed outcomes and interval bounds (e.g., transf=exp). Defaults to FALSE, which means that no transformation is used (any additional arguments needed for the function specified here can be passed via ...).
...	other arguments.

Value

The print.escalc function formats and prints the data frame, so that the observed outcomes and sampling variances are rounded (to the number of digits specified).

The summary.escalc function creates an object of class c("summary.escalc","data.frame"). The object is a data frame containing the original data (if append=TRUE) and the following components:

yi	observed outcomes or effect size estimates (transformed if transf is not FALSE).
vi	corresponding (estimated) sampling variances.
sei	standard errors of the observed outcomes or effect size estimates.
zi	test statistics for testing $H_0 : \theta_i = 0$ (i.e., yi/sei).
ci.lb	lower confidence interval bounds (transformed if transf is not FALSE).
ci.ub	upper confidence interval bounds (transformed if transf is not FALSE).

Note that the actual variable names above depend on the out.names (and var.names) arguments. If the data frame already contains variables with names as specified by the out.names argument, the values for these variables will be overwritten when replace=TRUE (which is the default). By setting replace=FALSE, only values that are NA will be replaced.

The print.summary.escalc function formats and prints the data frame.

Note

If some transformation function has been specified for the transf argument, then yi, ci.lb, and ci.ub will be transformed accordingly. However, vi and sei then still reflect the sampling variances and standard errors of the untransformed values.

The summary.escalc function computes level % Wald-type confidence intervals, which may or may not be the most accurate method for computing confidence intervals for the chosen outcome or effect size measure.

Author(s)

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

References

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

See Also

escalc

Examples

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

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

### apply summary function
summary(dat)
summary(dat, transf=exp)
```

---

print.fsn *Print Method for fsn Objects*

Description

Print method for objects of class "fsn".

Usage

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

Arguments

x	an object of class "fsn".
digits	integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
...	other arguments.

Details

The output shows the results from the fail-safe N calculation.

Value

The function does not return an object.

Author(s)

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

References

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

See Also

fsn

`print.list.rma` *Print Method for list.rma Objects*

Description

Print method for objects of class `"list.rma"`.

Usage

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

Arguments

`x`	an object of class `"list.rma"`.
`digits`	integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
`...`	other arguments.

Value

See the documentation of the function that creates the `"list.rma"` object for details on what is printed.

Author(s)

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

References

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

print.permutest.rma.uni

Print Method for permutest.rma.uni Objects

Description

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

Usage

```
## S3 method for class 'permutest.rma.uni'
print(x, digits=x$digits, signif.legend=TRUE, ...)
```

Arguments

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

Details

The output includes:

- the results of the omnibus test of the coefficients in the model. Suppressed if the model includes only one coefficient (e.g., only an intercept, like in the fixed- and random-effects model). The p-value based on the permutation test is indicated by p-val*.
- a table with the estimated coefficients, corresponding standard errors, test statistics, p-values, and confidence interval bounds. The p-values based on the permutation test are indicated by pval*.

Value

The function does not return an object.

Author(s)

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

References

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

See Also

permutest.rma.uni

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

Description

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

Usage

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

Arguments

x	an object of class "predict.rma".
digits	integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
...	other arguments.

Value

See the documentation of the predict.rma function for details on what is printed.

Author(s)

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

References

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

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

Description

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

Usage

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

Arguments

`x`	an object of class `"ranktest.rma"`.
`digits`	integer specifying the number of decimal places to which the printed results should be rounded (the default is to take the value from the object).
`...`	other arguments.

Details

The output includes:

- the estimated value of Kendall's tau rank correlation coefficient
- the corresponding p-value for the test that the true tau is equal to zero

Value

The function does not return an object.

Author(s)

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

References

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

See Also

`ranktest.rma`

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

Description

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

Usage

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

Arguments

x	an object of class "regtest.rma".
digits	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).
ret.fit	logical indicating whether the full results from the fitted model should also be returned.
...	other arguments.

Details

The output includes:

- the model used for the regression test
- the predictor used for the regression test
- the results from the fitted model (only when ret.fit=TRUE)
- the 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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

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

See Also

regtest.rma

---

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

Description

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

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

Arguments

x	an object of class "rma.uni", "rma.mh", "rma.peto", "rma.glmm", or "summary.rma" (for print).
object	an object of class "rma" (for summary).
digits	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 (default is FALSE for print and TRUE for summary).
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 setting showfit=TRUE or by default for summary).
- the amount of (residual) heterogeneity in the random/mixed-effects model (i.e., the estimate of τ^2 and its square root). Suppressed for fixed-effects models. The (asymptotic) standard error of the estimate of τ^2 is also provided (where possible).

- 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 amount of study-level variability (only for objects of class `"rma.glmm"` and when using a model that models study level differences as a random effect).
- the results of the test for (residual) heterogeneity. This is the usual Q-test for heterogeneity when not including moderators in the model and the QE-test for residual heterogeneity when moderators are included. For objects of class `"rma.glmm"`, the results from a likelihood ratio test are provided.
- the results of the omnibus (Wald-type) test of the coefficients in the model (the indices of the coefficients tested are also indicated). Suppressed if the model includes only one coefficient (e.g., only an intercept, like in the fixed- and random-effects model).
- a table with the estimated coefficients, corresponding standard errors, test statistics, p-values, and confidence interval bounds.

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 `print` functions do not return an object. The `summary` function returns the object passed to it (with additional class `"summary.rma"`).

Author(s)

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

References

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

See Also

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

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, label=FALSE, offset=0.3, ...)
## S3 method for class 'rma.mh'
qqnorm(y, type="rstandard", pch=19, label=FALSE, offset=0.3, ...)
## S3 method for class 'rma.peto'
qqnorm(y, type="rstandard", pch=19, label=FALSE, offset=0.3, ...)
## S3 method for class 'rma.glmm'
qqnorm(y, ...)
```

Arguments

y	an object of class "rma.uni", "rma.mh", or "rma.peto". The method is not yet implemented for objects of class "rma.glmm".
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).
label	argument to control the labeling of the points (default is FALSE). See 'Details'.
offset	argument to control the distance between the points and the corresponding labels (default is 0.3).
...	other arguments.

Details

The plot shows the theoretical quantiles of a normal distribution on the horizontal axis against the observed quantiles for either the standardized residuals (`type="rstandard"`, the default) or the externally standardized residuals (`type="rstudent"`) on the vertical axis (see `residuals.rma` for details on the definition of these residual types).

For reference, a line is added to the plot with slope of 1, going through the (0,0) point.

For objects of class `"rma.uni"`, it is also possible to add a pseudo confidence envelope to the plot. The envelope is created based on the quantiles of sets of pseudo residuals simulated from the given model (for details, see Cook & Weisberg, 1982). The number of sets simulated can be controlled with the `reps` argument. When `smooth=TRUE`, the simulated bounds are smoothed with Friedman's SuperSmoother (see `supsmu`). The `bass` argument can be set to a number between 0 and 10, with higher numbers indicating increasing smoothness. If `bonferroni=TRUE`, the envelope bounds are Bonferroni corrected, so that the envelope can be regarded as a confidence region for all k residuals simultaneously. The default however is `bonferroni=FALSE`, which makes the plot more sensitive to deviations from normality.

With the `label` argument, one can control whether points in the plot will be labeled (e.g., to identify outliers). If `label="all"`, all points in the plot will be labeled. If `label="out"`, points falling outside of the confidence envelope will be labeled (only available for objects of class `"rma.uni"`). Finally, one can also set this argument to a numeric value (between 1 and k), indicating how many of the most extreme points should be labeled (for example, with `label=1` only the most extreme point would be labeled, while with `label=3`, the most extreme, and the second and third most extreme points would be labeled). With the `offset` argument, the distance between the labels and the corresponding points can be controlled.

Value

A list with components:

`x`	the x coordinates of the points that were plotted.
`y`	the y coordinates of the points that were plotted.

Author(s)

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

References

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

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

Wang, M. C., & Bushman, B. J. (1998). Using the normal quantile plot to explore meta-analytic data sets. *Psychological Methods*, **3**, 46–54.

See Also

rma.uni, rma.mh, rma.peto, rma.glmm

Examples

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

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

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

---

radial *Radial (Galbraith) Plots*

Description

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

Usage

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

Arguments

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

Details

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

Author(s)

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

References

Galbraith, R. F. (1988). Graphical display of estimates having differing standard errors. *Technometrics*, **30**, 271–281.

Galbraith, R. F. (1988). A note on graphical presentation of estimated odds ratios from several clinical trials. *Statistics in Medicine*, **7**, 889–894.

Galbraith, R. F (1994). Some applications of radial plots. *Journal of the American Statistical Association*, **89**, 1232–1242.

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

See Also

radial.rma

`radial.rma` *Radial (Galbraith) Plots for rma Objects*

Description

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

Usage

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

Arguments

`x`	an object of class `"rma"`.
`center`	logical to indicate whether the plot should be centered horizontally at the model estimate (default is `FALSE`).
`xlim`	x axis limits. If unspecified, the function tries to set the x axis limits to some sensible values.
`zlim`	z axis limits. If unspecified, the function tries to set the z axis limits to some sensible values (note that the z axis limits are the actual vertical limit of the plotting region).
`xlab`	title for the x axis. If unspecified, the function tries to set an appropriate axis title.
`zlab`	title for the z axis. If unspecified, the function tries to set an appropriate axis title.
`atz`	position for the z axis tick marks and labels. If unspecified, these values are set by the function.

aty	position for the y axis tick marks and labels. If unspecified, these values are set by the function.
steps	the number of tick marks for the y axis (default is 7). Ignored when argument aty is used.
level	numerical value between 0 and 100 specifying the level of the z axis error region (the default is to take the value from the object).
digits	integer specifying the number of decimal places to which the tick mark labels of the y axis should be rounded (default is 2).
back	color of the z axis error region. Set to NA to suppress shading of the region.
transf	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 specifying the number of line segments to use when drawing the y axis and confidence interval arcs (default is 100).
cex	optional character and symbol expansion factor. If unspecified, the function tries to set this to a sensible value.
...	other arguments.

Details

For a fixed-effects model, the plot shows the inverse of the standard errors on the horizontal axis against the individual observed effect sizes or outcomes standardized by their corresponding standard errors on the vertical axis. Since the vertical axis corresponds to standardized values, it is referred to as the z axis within this function. On the right hand side of the plot, an arc is drawn (referred to as the y axis within this function) corresponding to the individual observed effect sizes or outcomes. A line projected from (0,0) through a particular point within the plot onto this arc indicates the value of the individual observed effect size or outcome for that point.

For a random-effects model, the function uses $1/\sqrt{v_i + \tau^2}$ for the horizontal axis, where v_i is the sampling variance of the observed effect size or outcome and τ^2 is the amount of heterogeneity as estimated based on the model. For the z axis, $\sqrt{v_i + \tau^2}$ is used to standardize the individual observed effect sizes or outcomes.

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

Author(s)

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

References

Galbraith, R. F. (1988). Graphical display of estimates having differing standard errors. *Technometrics*, **30**, 271–281.

Galbraith, R. F. (1988). A note on graphical presentation of estimated odds ratios from several clinical trials. *Statistics in Medicine*, **7**, 889–894.

Galbraith, R. F (1994). Some applications of radial plots. *Journal of the American Statistical Association*, **89**, 1232–1242.

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

See Also

rma.uni, rma.mh, rma.peto, rma.glmm

Examples

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

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

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

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

---

ranktest *Rank Correlation Test for Funnel Plot Asymmetry*

Description

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

Usage

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

Arguments

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

Details

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

Author(s)

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

References

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

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

See Also

ranktest.rma

---

ranktest.rma *Rank Correlation Test for Funnel Plot Asymmetry for rma Objects*

Description

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

Usage

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

Arguments

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

Details

The function carries out the rank correlation test as described by Begg and Mazumdar (1994). The test can be used to examine whether the observed outcomes and the corresponding sampling variances are correlated. A high correlation would indicate that the funnel plot is asymmetric, which may be a result of publication bias.

Value

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

tau	the estimated value of Kendall's tau rank correlation coefficient
pval	the corresponding p-value for the test that the true tau is equal to zero

The results are 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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

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

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

See Also

ranktest, regtest

Examples

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

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

### 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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

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

Egger, M., Davey Smith, G., Schneider, M., & Minder, C. (1997). Bias in meta-analysis detected by a simple, graphical test. *British Medical Journal*, **315**, 629–634.

Harbord, R. M., Egger, M., & Sterne, J. A. C. (2006). A modified test for small-study effects in meta-analyses of controlled trials with binary endpoints. *Statistics in Medicine*, **25**, 3443–3457.

Macaskill, P., Walter, S. D., & Irwig, L. (2001). A comparison of methods to detect publication bias in meta-analysis. *Statistics in Medicine*, **20**, 641–654.

Moreno, S. G., Sutton, A. J., Ades, A. E., Stanley, T. D., Abrams, K. R., Peters, J. L., & Cooper, N. J. (2009). Assessment of regression-based methods to adjust for publication bias through a comprehensive simulation study. *BMC Medical Research Methodology*, **9**:2.

Peters, J. L., Sutton, A. J., Jones, D. R., Abrams, K. R., & Rushton, L. (2006). Comparison of two methods to detect publication bias in meta-analysis. *Journal of the American Medical Association*, **295**, 676–680.

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

Sterne, J. A. C., & Egger, M. (2005). Regression methods to detect publication and other bias in meta-analysis. In H. R. Rothstein, A. J. Sutton, & M. Borenstein (Eds.) *Publication bias in meta-analysis: Prevention, assessment, and adjustments* (pp. 99–110). Chichester, England: Wiley.

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

See Also

regtest.rma

---

regtest.rma *Regression Tests for Funnel Plot Asymmetry for rma Objects*

Description

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

Usage

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

Arguments

x	an object of class "rma".
model	either "rma" or "lm" to indicate the type of model to use for the regression test. See 'Details'.
predictor	either "sei" "vi", "ni", or "ninv" to indicate the type of independent variable to use for the regression test. See 'Details'.
ret.fit	logical indicating whether the full results from the fitted model should also be returned.
...	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 sample size, and the inverse of the 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, which in turn 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/mixed-effects model, one should first fit a random-effects model with the `rma` function and then use the `regtest.rma` function on the fitted model object.

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 sample size, and `predictor="ninv"` for the inverse of the sample size. For `predictor="ni"` and `predictor="ninv"`, the object `x` must contain the information about the sample sizes. This will be the case when `measure` was *not* equal to `"GEN"` or the `ni` values were explicitly specified 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:

`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).
`fit`	the full results from the fitted model.

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

Note

The classical "Egger test" is obtained by setting `model="lm"` and `predictor="sei"`. For the random/mixed-effects version of the Egger test, one should first fit a random-effects model to the data and then set `model="rma"` and `predictor="sei"` when using the `regtest.rma` function. See Sterne and Egger (2005) for details on these two types of models/tests.

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, which in turn may be an indication of publication bias. However, it is important to keep in mind that there can be other reasons besides publication bias that could lead to asymmetry in the funnel plot.

Author(s)

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

References

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

Egger, M., Davey Smith, G., Schneider, M., & Minder, C. (1997). Bias in meta-analysis detected by a simple, graphical test. *British Medical Journal*, **315**, 629–634.

Harbord, R. M., Egger, M., & Sterne, J. A. C. (2006). A modified test for small-study effects in meta-analyses of controlled trials with binary endpoints. *Statistics in Medicine*, **25**, 3443–3457.

Macaskill, P., Walter, S. D., & Irwig, L. (2001). A comparison of methods to detect publication bias in meta-analysis. *Statistics in Medicine*, **20**, 641–654.

Moreno, S. G., Sutton, A. J., Ades, A. E., Stanley, T. D., Abrams, K. R., Peters, J. L., & Cooper, N. J. (2009). Assessment of regression-based methods to adjust for publication bias through a comprehensive simulation study. *BMC Medical Research Methodology*, **9**:2.

Peters, J. L., Sutton, A. J., Jones, D. R., Abrams, K. R., & Rushton, L. (2006). Comparison of two methods to detect publication bias in meta-analysis. *Journal of the American Medical Association*, **295**, 676–680.

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

Sterne, J. A. C., & Egger, M. (2005). Regression methods to detect publication and other bias in meta-analysis. In H. R. Rothstein, A. J. Sutton, & M. Borenstein (Eds.) *Publication bias in meta-analysis: Prevention, assessment, and adjustments* (pp. 99–110). Chichester, England: Wiley.

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

See Also

regtest, ranktest

Examples

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

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

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

### classical Egger test
regtest(res, model="lm")
```

```
### random/mixed-effects version of the Egger test
regtest(res)

### more examples
res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat)

regtest(res, model="lm", predictor="ni")
regtest(res, model="lm", predictor="ninv")

regtest(res, model="rma", predictor="ni")
regtest(res, model="rma", predictor="ninv")

### testing for asymmetry after accounting for the influence of a moderator
res <- rma(measure="RR", ai=tpos, bi=tneg, ci=cpos, di=cneg,
           data=dat, mods = ~ ablat, method="REML")
regtest(res, model="lm")
regtest(res)
```

`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	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@metafor-project.org>
package homepage: http://www.metafor-project.org/
author homepage: http://www.wvbauer.com/

References

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

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

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

See Also

rma.uni, rma.mh, rma.peto, rma.glmm, influence.rma.uni

Examples

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

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

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

---

rma.glmm *Meta-Analysis via the Generalized Linear (Mixed-Effects) Model*

Description

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

Usage

```
rma.glmm(ai, bi, ci, di, n1i, n2i, x1i, x2i, t1i, t2i, xi, mi, ti, ni,
         mods, measure, intercept=TRUE, data, slab, subset,
         add=1/2, to="only0", drop00=TRUE, vtype="LS",
         model="UM.FS", method="ML", tdist=FALSE,
         level=95, digits=4, btt, nAGQ=1, verbose=FALSE, control)
```

Arguments

ai	see below and the documentation of the escalc function for more details.
bi	see below and the documentation of the escalc function for more details.
ci	see below and the documentation of the escalc function for more details.
di	see below and the documentation of the escalc function for more details.
n1i	see below and the documentation of the escalc function for more details.
n2i	see below and the documentation of the escalc function for more details.
x1i	see below and the documentation of the escalc function for more details.
x2i	see below and the documentation of the escalc function for more details.
t1i	see below and the documentation of the escalc function for more details.
t2i	see below and the documentation of the escalc function for more details.
xi	see below and the documentation of the escalc function for more details.
mi	see below and the documentation of the escalc function for more details.
ti	see below and the documentation of the escalc function for more details.
ni	see below and the documentation of the escalc function for more details.
mods	optional argument to include one or more moderators in the model. A single moderator can be given as a vector of length k specifying the values of the moderator. Multiple moderators are specified by giving a matrix with k rows and p' columns. Alternatively, a model formula can be used to specify the model. See 'Details'.
measure	character string indicating the outcome measure to use for the meta-analysis. Possible options are the odds ratio ("OR"), the incidence rate ratio ("IRR"), the logit transformed proportion ("PLO"), or the log transformed incidence rate ("IRLN").
intercept	logical, indicating whether an intercept term should be added to the model (default is TRUE).
data	optional data frame containing the data supplied to the function.
slab	optional vector with unique labels for the k studies.
subset	optional vector indicating the subset of studies that should be used for the analysis. This can be a logical vector of length k or a numeric vector indicating the indices of the observations to include.
add	non-negative number indicating the amount to add to zero cells, counts, or frequencies when calculating the individual outcomes. See below and the documentation of the escalc function for more details.
to	character string indicating when the values under add should be added (either "only0", "all", "if0all", or "none"). See below and the documentation of the escalc function for more details.
drop00	logical indicating whether studies with no cases/events (or only cases) in both groups should be dropped. See the documentation of the escalc function for more details.

vtype	character string indicating the type of sampling variances to calculate when calculating the individual outcomes. See below and the documentation of the escalc function for more details.
model	character string specifying the general model type to use for the analysis (either "UM.FS" (the default), "UM.RS", "CM.EL", or "CM.AL"). See 'Details'.
method	character string 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="ML" (the default). See 'Details'.
tdist	logical specifying whether test statistics and confidence intervals should be based on the normal (when FALSE, the default) or the t-distribution (when TRUE). See 'Details'.
level	numerical value between 0 and 100 specifying the confidence interval level (default is 95).
digits	integer specifying the number of decimal places to which the printed results should be rounded (default is 4).
btt	optional vector of indices specifying which coefficients to include in the omnibus test of moderators. See 'Details'.
nAGQ	positive integer specifying the number of points per axis for evaluating the adaptive Gauss-Hermite approximation to the log likelihood. This defaults to 1, corresponding to the Laplacian approximation. Values greater than 1 produce greater accuracy in the evaluation of the log likelihood at the expense of speed. See 'Note'.
verbose	logical indicating whether output should be generated for the fitting algorithms used (default is FALSE). See 'Note'.
control	optional list of control values for the 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 the following effect size or outcome measures:

- measure="OR" for odds ratios (analyzed in log units)
- measure="IRR" for incidence rate ratios (analyzed in log units)
- measure="PLO" for logit transformed proportions (i.e., log odds)
- measure="IRLN" for log transformed incidence rates.

The escalc function describes the data/arguments that should be specified/used for these measures.

Specifying the Model

A variety of general model types are available when analyzing 2x2 table data (i.e., when measure="OR") or two-group event count data (i.e., when measure="IRR").

- model="UM.FS" for an unconditional generalized linear mixed-effects model with fixed study effects

- `model="UM.RS"` for an unconditional generalized linear mixed-effects model with random study effects
- `model="CM.AL"` for a conditional generalized linear mixed-effects model (approximate likelihood)
- `model="CM.EL"` for a conditional generalized linear mixed-effects model (exact likelihood).

For `measure="OR"`, models `"UM.FS"` and `"UM.RS"` are essentially (mixed-effects) logistic regression models, while for `measure="IRR"`, these models are (mixed-effects) Poisson regression models. A choice must be made on how to model study level variability (i.e., differences in outcomes across studies irrespective of group membership). One can choose between using fixed study effects (which means that k dummy variables are added to the model) or random study effects (which means adding a random effect at the study level).

The conditional model (`model="CM.EL"`) avoids having to model study level variability by conditioning on the total numbers of cases/events in each study. For `measure="OR"`, this leads to a non-central hypergeometric distribution for the data within each study and the corresponding model is then a (mixed-effects) conditional logistic model. Fitting this model can be difficult and computationally expensive. When the number of cases in each study is small relative to the group sizes, one can approximate the exact likelihood by a binomial distribution, which leads to a regular (mixed-effects) logistic regression model (`model="CM.AL"`). For `measure="IRR"`, the conditional model leads directly to a binomial distribution for the data within each study and the resulting model is again a (mixed-effects) logistic regression model (no approximate likelihood model is needed here).

When analyzing proportions (i.e., when `measure="PLO"`) or incidence rates (i.e., when `measure="IRLN"`) of individual groups, the model type is always a (mixed-effects) logistic or Poisson regression model, respectively (i.e., the `model` argument is not relevant here).

Aside from choosing the general model type, one has to decide whether to fit a fixed- or random-effects model to the data. A *fixed-effects model* is fitted by setting `method="FE"`. A *random-effects model* is fitted by setting `method="ML"` (the default). Note that random-effects models with dichotomous data are often referred to as 'binomial-normal' models in the meta-analytic literature. Analogously, for event count data, such models could be referred to as 'Poisson-normal' models.

One or more moderators can be included in all of these models via the `mods` argument. A single moderator can be given as a (row or column) vector of length k specifying the values of the moderator. Multiple moderators are specified by giving an appropriate design matrix with k rows and p' columns (e.g., using `mods = cbind(mod1, mod2, mod3)`, where `mod1`, `mod2`, and `mod3` correspond to the names of the variables for the three moderator variables). The intercept is included in the model by default unless `intercept=FALSE`.

Alternatively, one can use standard `formula` syntax to specify the model. In this case, the `mods` argument should be set equal to a one-sided formula of the form `mods = ~ model` (e.g., `mods = ~ mod1 + mod2 + mod3`). Interactions, polynomial terms, and factors can be easily added to the model in this manner. When specifying a model formula via the `mods` argument, the `intercept` argument is ignored. Instead, the inclusion/exclusion of the intercept term is controlled by the specified formula (e.g., `mods = ~ mod1 + mod2 + mod3 - 1` would lead to the removal of the intercept term). With moderators, a *fixed-effects with moderators model* is then fitted by setting `method="FE"`, while a *mixed-effects model* is fitted by setting `method="ML"`.

Fixed-, Saturated-, and Random/Mixed-Effects Models

When fitting a particular model, actually up to three different models are fitted within the function:

- the fixed-effects model (i.e., where τ^2 is set to 0),

- the saturated model (i.e., the model with a deviance of 0), and
- the random/mixed-effects model (i.e., where τ^2 is estimated) (only if `method="ML"`).

The saturated model is obtained by adding as many dummy variables to the model as needed so that the model deviance is equal to zero. Even when `method="ML"`, the fixed-effects and saturated models are fitted, as they are used to compute the test statistics for the Wald-type and likelihood ratio tests for (residual) heterogeneity (see below).

Omnibus Test of Parameters

For models including moderators, an omnibus test of all the model coefficients is conducted that excludes the intercept (the first coefficient) if it is included in the model. If no intercept is included in the model, then the omnibus test includes all of the coefficients in the model including the first. Alternatively, one can manually specify the indices of the coefficients to test via the `btt` argument. For example, use `btt=c(3,4)` to only include the third and fourth coefficient from the model in the test (if an intercept is included in the model, then it corresponds to the first coefficient in the model).

Categorical Moderators

Categorical moderator variables can be included in the model in the same way that appropriately (dummy) coded categorical independent variables can be included in linear models. One can either do the dummy coding manually or use a model formula together with the `factor` function to let R handle the coding automatically.

Tests and Confidence Intervals

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). As an alternative, one can set `tdist=TRUE`, which slightly mimics the Knapp and Hartung (2003) method by using a t-distribution with $k - p$ degrees of freedom for tests of individual coefficients and confidence intervals and an F-distribution with m and $k - p$ degrees of freedom (p being the total number of model coefficients including the intercept if it is present) for the omnibus test statistic.

Tests for (Residual) Heterogeneity

Two different tests for (residual) heterogeneity are automatically carried out by the function. The first is a Wald-type test, which tests the coefficients corresponding to the dummy variables added in the saturated model for significance. The second is a likelihood ratio test, which tests the same set of coefficients, but does so by comparing the deviance of the fixed-effects and the saturated model. These two tests are not identical for the types of models fitted by the `rma.glmm` function and may even lead to conflicting conclusions.

Individual Outcomes

The various models do not require the calculation of the individual outcome values and directly make use of the table/event counts. Zero cells/events are not a problem (except in extreme cases, such as when one of the two outcomes never occurs or when there are no events in any of the studies). Therefore, it is unnecessary to add some constant to the cell counts (or the number of events) when there are zero cells/events. However, for plotting and various other functions, it is necessary to calculate the individual outcome values for the k studies. Here, zero cells/events can be problematic, so adding a constant value to the cell counts (or the number of events) ensures that all k values can be calculated. The `add` and `to` arguments are used to specify what value should be added to the cell frequencies (or the number of events) and under what circumstances when calculating the individual outcome values. The documentation of the `escalc` function explains how

the add and to arguments work. Note that drop00 is set to TRUE by default, since studies where ai=ci=0 or bi=di=0 or studies where x1i=x2i=0 are uninformative about the size of the effect (the counts for such studies are set to NA).

Value

An object of class c("rma.glmm","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".
sigma2	estimated amount of study level variability (only for model="UM.RS").
k	number of studies included in the model.
p	number of coefficients in the model (including the intercept).
m	number of coefficients included in the omnibus test of coefficients.
QE.Wld	Wald-type test statistic for the test of (residual) heterogeneity.
QEp.Wld	p-value for the Wald-type test of (residual) heterogeneity.
QE.LRT	likelihood ratio test statistic for the test of (residual) heterogeneity.
QEp.LRT	p-value for the likelihood ratio 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 is an intercept-only model.
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.
...	some additional elements/values.

The results of the fitted model are neatly formated and printed with the print.rma.uni function. If fit statistics should also be given, use summary.rma (or use the fitstats.rma function to extract them).

Note

Fitting the various types of models requires several different iterative algorithms:

- For `model="UM.FS"` and `model="CM.AL"`, iteratively reweighted least squares (IWLS) as implemented in the `glm` function is used for fitting the fixed-effects and the saturated models. For `method="ML"`, adaptive Gauss-Hermite quadrature as implemented in the `lmer` function is used. The same applies when `model="CM.EL"` is used in combination with `measure="IRR"` or when `measure="PLO"` or `measure="IRLN"` (regardless of which general model type is then specified).
- For `model="UM.RS"`, adaptive Gauss-Hermite quadrature as implemented in the `lmer` function is used to fit all of the models.
- For `model="CM.EL"` and `measure="OR"`, the quasi-Newton method (`"BFGS"`) as implemented in the `optim` function is used for fitting the fixed-effects and the saturated models. For `method="ML"`, the same algorithm is used, together with adaptive quadrature as implemented in the `integrate` function (for the integration over the density of the non-central hypergeometric distribution).

When `model="CM.EL"` and `measure="OR"`, actually `model="CM.AL"` is first used to obtain starting values for `optim`, so either 4 (if `method="FE"`) or 6 (if `method="ML"`) models need to be fitted in total.

Some important control parameters for the various algorithms can be adjusted via the `control` argument. In particular,

- for `glm`, the `epsilon` (default is `1e-8`) and `maxit` (default is `25`) arguments,
- for `lmer`, the `maxIter` (default is `300`) and `maxFN` (default is `900`) arguments,
- for `optim`, the `method` (default is `"BFGS"`) and `reltol` (default is `1e-8`) arguments,
- for `integrate`, the `rel.tol` (default is `1e-8`) and `subdivisions` (default is `100`) arguments.

Also, for `lmer`, the `nAGQ` argument is used to specify the number of quadrature points. The default is 1, which corresponds to the Laplacian approximation. Values greater than 1 produce greater accuracy in the evaluation of the log likelihood at the expense of speed.

Information on the evolution of the various algorithms is obtained by setting `verbose=TRUE` or with `control=list(verbose=TRUE)`.

For `model="CM.EL"` and `measure="OR"`, optimization involves repeated calculation of the density of the non-central hypergeometric distribution. When `method="ML"`, this also requires integration over the same density. This is currently implemented in a rather brute-force manner and may not be numerically stable, especially when models with moderators are fitted. Stability can be improved by scaling the moderators in a similar manner (i.e., don't use a moderator that is coded 0 and 1, while another uses values in the 1000s). Sensitivity analyses are highly recommended here, to ensure that the results do not depend on the scaling of the moderators.

Author(s)

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

References

Agresti, A. (2002). *Categorical data analysis* (2nd. ed). Hoboken, NJ: Wiley.

Bagos, P. G., & Nikolopoulos, G. K. (2009). Mixed-effects Poisson regression models for meta-analysis of follow-up studies with constant or varying durations. *The International Journal of Biostatistics*, **5**(1), article 21.

van Houwelingen, H. C., Zwinderman, K. H., & Stijnen, T. (1993). A bivariate approach to meta-analysis. *Statistics in Medicine*, **12**, 2273–2284.

Stijnen, T., Hamza, T. H., & Ozdemir, P. (2010). Random effects meta-analysis of event outcome in the framework of the generalized linear mixed model with applications in sparse data. *Statistics in Medicine*, **29**, 3046–3067.

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

See Also

rma.uni, rma.mh, rma.peto

Examples

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

### random-effects model using rma.uni() (standard random-effects model analysis)
rma(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, method="REML")

### random-effects models using rma.glmm() (require lme4 package to be installed)

### unconditional model with fixed study effects
rma.glmm(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, model="UM.FS")

### unconditional model with random study effects
rma.glmm(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, model="UM.RS")

### conditional model with approximate likelihood
rma.glmm(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, model="CM.AL")

### conditional model with exact likelihood (fitting this model is slow)
## Not run:
rma.glmm(measure="OR", ai=tpos, bi=tneg, ci=cpos, di=cneg, data=dat.bcg, model="CM.EL")

## End(Not run)
```

---

rma.mh *Meta-Analysis via the Mantel-Haenszel Method*

Description

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

Usage

```
rma.mh(ai, bi, ci, di, n1i, n2i, x1i, x2i, t1i, t2i,
       measure="OR", data, slab, subset,
       add=1/2, to="only0", drop00=TRUE,
       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.
x1i	vector to specify the number of events (first group). See below and the documentation of the escalc function for more details.
x2i	vector to specify the number of events (second group). See below and the documentation of the escalc function for more details.
t1i	vector to specify the total person-times (first group). See below and the documentation of the escalc function for more details.
t2i	vector to specify the total person-times (second group). See below and the documentation of the escalc function for more details.
measure	character string indicating the outcome measure to use for the meta-analysis. Possible options are the odds ratio ("OR"), the relative risk ("RR"), the risk difference ("RD"), or the incidence rate ratio ("IRR").
data	optional data frame containing the data supplied to the function.
slab	optional vector with unique labels for the k studies.
subset	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	non-negative number indicating the amount to add to zero cells, counts, or frequencies when calculating the individual outcomes. Can also be a vector of two numbers, where the first number is used in the calculation of the individual outcomes and the second number is used when applying the Mantel-Haenszel method. See below and the documentation of the escalc function for more details.

to	character string indicating when the values under add should be added (either "only0", "all", "if0all", or "none"). Can also be a character vector, where the first string again applies when calculating the individual outcomes and the second string when applying the Mantel-Haenszel method. See below and the documentation of the escalc function for more details.
drop00	logical indicating whether studies with no cases/events (or only cases) in both groups should be dropped when calculating the observed outcomes of the individual studies (the outcomes for such studies are set to NA). See below and the documentation of the escalc function for more details.
level	numerical value between 0 and 100 specifying the confidence interval level (default is 95).
digits	integer specifying the number of decimal places to which the printed results should be rounded (default is 4).

Details

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

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

where ai, bi, ci, and di denote the cell frequencies and n1i and n2i the row totals. For example, in a set of randomized clinical trials (RCTs) or cohort studies, group 1 and group 2 may refer to the treatment (exposed) and placebo/control (not exposed) group, with outcome 1 denoting some event of interest (e.g., death) and outcome 2 its complement. In a set of case-control studies, group 1 and group 2 may refer to the group of cases and the group of controls, with outcome 1 denoting, for example, exposure to some risk factor and outcome 2 non-exposure. For these outcome measures, one needs to specify either ai, bi, ci, and di or alternatively ai, ci, n1i, and n2i.

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

	events	person-time
group 1	x1i	t1i
group 2	x2i	t2i

where x1i and x2i denote the number of events in the first and the second group, respectively, and t1i and t2i the corresponding total person-times at risk.

An approach for aggregating table data of these types was suggested by Mantel and Haenszel (1959) and later extended by various authors (see references). The Mantel-Haenszel method provides a weighted estimate under a fixed-effects model. The method is particularly advantageous when aggregating a large number of studies with small sample sizes (the so-called sparse data or increasing strata case). When analyzing odds ratios, the Cochran-Mantel-Haenszel test (Cochran, 1954; Mantel & Haenszel, 1959) and Tarone's test for heterogeneity (Tarone, 1985) are also provided. When

analyzing odds ratios, relative risks, or incidence rate ratios, the printed results are given both in terms of the log and the raw units (for easier interpretation).

The Mantel-Haenszel method itself does not require the calculation of the individual outcome values and directly makes use of the table/event counts. Zero cells/events are not a problem (except in extreme cases, such as when one of the two outcomes never occurs in any of the 2x2 tables or when there are no events for one of the two groups in any of the tables). Therefore, it is unnecessary to add some constant to the cell counts (or the number of events) when there are zero cells/events. However, for plotting and various other functions, it is necessary to calculate the individual outcome values for the k studies. Here, zero cells/events can be problematic, so adding a constant value to the cell counts (or the number of events) ensures that all k values can be calculated. The `add` and `to` arguments are used to specify what value should be added to the 2x2 cell frequencies (or the number of events) and under what circumstances when calculating the individual outcome values and when applying the Mantel-Haenszel method. The documentation of the `escalc` function explains how the `add` and `to` arguments work. If only one value for these arguments is specified, then these values are used when calculating the individual outcomes and no adjustment to the cell counts (or the number of events) is made when applying the Mantel-Haenszel method. Alternatively, when specifying two values for these arguments, the first value applies when calculating the individual outcomes and the second value when applying the Mantel-Haenszel method. Note that `drop00` is set to `TRUE` by default, since studies where `ai=ci=0` or `bi=di=0` or studies where `x1i=x2i=0` are also automatically 'dropped' 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 relative risk, 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 fit statistics should also be given, use `summary.rma` (or use the `fitstats.rma` function to extract them).

The `residuals.rma`, `rstandard.rma.mh`, and `rstudent.rma.mh` functions extract raw and standardized residuals. Leave-one-out diagnostics can be obtained with `leave1out.rma.mh`.

Forest, funnel, radial, and L'Abbe plots can be obtained with `forest.rma`, `funnel.rma`, `radial.rma`, and `labbe.rma`. The `qqnorm.rma.mh` function provides normal QQ plots of the standardized residuals. One can also just call `plot.rma.mh` on the fitted model object to obtain various plots at once.

A cumulative meta-analysis (i.e., adding one obervation at a time) can be obtained with `cumul.rma.mh`.

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

Author(s)

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

References

Cochran, W. G. (1954). Some methods for strengthening the common χ^2 tests. *Biometrics*, **10**, 417–451.

Greenland, S., & Robins, J. M. (1985). Estimation of a common effect parameter from sparse follow-up data. *Biometrics*, **41**, 55–68.

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.

Nurminen, M. (1981). Asymptotic efficiency of general noniterative estimators of common relative risk. *Biometrika*, **68**, 525–530.

Robins, J., Breslow, N., & Greenland, S. (1986). Estimators of the Mantel-Haenszel variance consistent in both sparse data and large-strata limiting models. *Biometrics*, **42**, 311–323.

Sato, T., Greenland, S., & Robins, J. M. (1989). On the variance estimator for the Mantel-Haenszel risk difference. *Biometrics*, **45**, 1323–1324.

Tarone, R. E. (1981). On summary estimators of relative risk. *Journal of Chronic Diseases*, **34**, 463–468.

Tarone, R. E. (1985). On heterogeneity tests based on efficient scores. *Biometrika*, **72**, 91–95.

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

See Also

`rma.uni`, `rma.glmm`, `rma.peto`

Examples

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

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

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

---

rma.peto *Meta-Analysis via Peto's Method*

Description

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

Usage

```
rma.peto(ai, bi, ci, di, n1i, n2i,
         data, slab, subset,
         add=1/2, to="only0", drop00=TRUE,
         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	optional data frame containing the data supplied to the function.
slab	optional vector with unique labels for the k studies.
subset	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	non-negative number indicating the amount to add to zero cells, counts, or frequencies when calculating the individual outcomes. Can also be a vector of two numbers, where the first number is used in the calculation of the individual outcomes and the second number is used when applying Peto's method. See below and the documentation of the escalc function for more details.
to	character string indicating when the values under add should be added (either "only0", "all", "if0all", or "none"). Can also be a character vector, where the first string again applies when calculating the individual outcomes and the second string when applying Peto's method. See below and the documentation of the escalc function for more details.
drop00	logical indicating whether studies with no cases (or only cases) in both groups should be dropped when calculating the observed outcomes of the individual studies (the outcomes for such studies are set to NA). See below and the documentation of the escalc function for more details.
level	numerical value between 0 and 100 specifying the confidence interval level (default is 95).
digits	integer specifying the number of decimal places to which the printed results should be rounded (default is 4).

Details

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

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

where ai, bi, ci, and di denote the cell frequencies and n1i and n2i the row totals. For example, in a set of randomized clinical trials (RCTs) or cohort studies, group 1 and group 2 may refer to the treatment (exposed) and placebo/control (not exposed) group, with outcome 1 denoting some event of interest (e.g., death) and outcome 2 its complement. In a set of case-control studies, group 1 and group 2 may refer to the group of cases and the group of controls, with outcome 1 denoting, for example, exposure to some risk factor and outcome 2 non-exposure.

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).

Peto's 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 unnecessary to add some constant to the cell counts when there are zero cells. However, for plotting and various other functions, it is necessary to calculate the individual (log) odds ratios for the k tables. Here, zero cells can be problematic, so adding a constant value to the cell counts ensures that all k values can be calculated. The add and to arguments are used to specify what value should be added to the 2x2 cell frequencies and under what circumstances when calculating the individual (log) odds ratios and when applying Peto's method. The documentation of the escalc function explains how the add and to arguments work. If only one value for these arguments is specified, then these values are

used when calculating the individual outcomes and no adjustment to the cell counts is made when applying Peto's method. Alternatively, when specifying two values for these arguments, the first value applies when calculating the individual outcomes and the second value when applying Peto's method. Note that `drop00` is set to `TRUE` by default, since studies where `ai=ci=0` or `bi=di=0` are also automatically 'dropped' 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 fit statistics should also be given, use `summary.rma` (or use the `fitstats.rma` function to extract them).

The `residuals.rma`, `rstandard.rma.peto`, and `rstudent.rma.peto` functions extract raw and standardized residuals. Leave-one-out diagnostics can be obtained with `leave1out.rma.peto`.

Forest, funnel, radial, and L'abbe plots can be obtained with `forest.rma`, `funnel.rma`, `radial.rma`, and `labbe.rma`. The `qqnorm.rma.peto` function provides normal QQ plots of the standardized residuals. One can also just call `plot.rma.peto` on the fitted model object to obtain various plots at once.

A cumulative meta-analysis (i.e., adding one obervation at a time) can be obtained with `cumul.rma.peto`.

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

Author(s)

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

References

Yusuf, S., Peto, R., Lewis, J., Collins, R., & Sleight, P. (1985). Beta blockade during and after myocardial infarction: An overview of the randomized trials. *Progress in Cardiovascular Disease*, **27**, 335–371.

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

See Also

rma.uni, rma.glmm, rma.mh

Examples

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

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

---

rma.uni *Meta-Analysis via the Linear (Mixed-Effects) Model*

Description

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

Usage

```
rma.uni(yi, vi, sei, weights, ai, bi, ci, di, n1i, n2i, x1i, x2i,
        t1i, t2i, m1i, m2i, sd1i, sd2i, xi, mi, ri, ti, sdi, ni, mods,
        measure="GEN", intercept=TRUE, data, slab, subset,
        add=1/2, to="only0", drop00=FALSE, vtype="LS",
        method="REML", weighted=TRUE, knha=FALSE,
        level=95, digits=4, btt, tau2, verbose=FALSE, control)
rma(yi, vi, sei, weights, ai, bi, ci, di, n1i, n2i, x1i, x2i,
        t1i, t2i, m1i, m2i, sd1i, sd2i, xi, mi, ri, ti, sdi, ni, mods,
        measure="GEN", intercept=TRUE, data, slab, subset,
        add=1/2, to="only0", drop00=FALSE, vtype="LS",
        method="REML", weighted=TRUE, knha=FALSE,
        level=95, digits=4, btt, tau2, verbose=FALSE, control)
```

Arguments

yi	vector of length k with the observed effect sizes or outcomes. See 'Details'.
vi	vector of length k with the corresponding sampling variances. See 'Details'.
sei	vector of length k with the corresponding standard errors. See 'Details'.
weights	vector of length k with the corresponding inverse sampling variances. See 'Details'.
ai	see below and the documentation of the escalc function for more details.
bi	see below and the documentation of the escalc function for more details.
ci	see below and the documentation of the escalc function for more details.
di	see below and the documentation of the escalc function for more details.
n1i	see below and the documentation of the escalc function for more details.
n2i	see below and the documentation of the escalc function for more details.
x1i	see below and the documentation of the escalc function for more details.
x2i	see below and the documentation of the escalc function for more details.
t1i	see below and the documentation of the escalc function for more details.
t2i	see below and the documentation of the escalc function for more details.
m1i	see below and the documentation of the escalc function for more details.
m2i	see below and the documentation of the escalc function for more details.
sd1i	see below and the documentation of the escalc function for more details.
sd2i	see below and the documentation of the escalc function for more details.
xi	see below and the documentation of the escalc function for more details.
mi	see below and the documentation of the escalc function for more details.
ri	see below and the documentation of the escalc function for more details.
ti	see below and the documentation of the escalc function for more details.
sdi	see below and the documentation of the escalc function for more details.
ni	see below and the documentation of the escalc function for more details.
mods	optional argument to include one or more moderators in the model. A single moderator can be given as a vector of length k specifying the values of the moderator. Multiple moderators are specified by giving a matrix with k rows and p' columns. Alternatively, a model formula can be used to specify the model. See 'Details'.
measure	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 specify the needed data via the appropriate arguments.
intercept	logical indicating whether an intercept term should be added to the model (default is TRUE).

`data`	optional data frame containing the data supplied to the function.
`slab`	optional vector with unique labels for the k studies.
`subset`	optional vector indicating the subset of studies that should be used for the analysis. This can be a logical vector of length k or a numeric vector indicating the indices of the observations to include.
`add`	see the documentation of the `escalc` function.
`to`	see the documentation of the `escalc` function.
`drop00`	see the documentation of the `escalc` function.
`vtype`	see the documentation of the `escalc` function.
`method`	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: `"DL"`, `"HE"`, `"SJ"`, `"ML"`, `"REML"`, `"EB"`, or `"HS"`. Default is `"REML"`. See 'Details'.
`weighted`	logical indicating whether weighted (default) or unweighted least squares should be used to fit the model.
`knha`	logical specifying whether the method by Knapp and Hartung (2003) should be used for adjusting test statistics and confidence intervals (default is `FALSE`). See 'Details'.
`level`	numerical value between 0 and 100 specifying the confidence interval level (default is 95).
`digits`	integer specifying the number of decimal places to which the printed results should be rounded (default is 4).
`btt`	optional vector of indices specifying which coefficients to include in the omnibus test of moderators. See 'Details'.
`tau2`	optional numerical value to specify the amount of (residual) heterogeneity in a random- or mixed-effects model (instead of estimating it). Useful for sensitivity analyses (e.g., for plotting results as a function of τ^2). When unspecified, the value of τ^2 is estimated from the data.
`verbose`	logical indicating whether output should be generated for the Fisher scoring algorithm (default is `FALSE`). See 'Note'.
`control`	optional list of control values for the iterative estimation algorithms. Defaults to an empty list, which means that default values are defined inside the function. See 'Note'.

Details

Specifying the Data

The function can be used in conjunction with any of the usual effect size or outcome measures used in meta-analyses (e.g., log odds ratios, log relative risks, risk differences, mean differences, standardized mean differences, raw correlation coefficients, correlation coefficients transformed with Fisher's r-to-z transformation, and so on). Simply specify the observed outcomes via the `yi` argument and the corresponding sampling variances via the `vi` argument (instead, one can specify

the standard errors, the square root of the sampling variances, via the `sei` argument, or the inverse of the sampling variances via the `weights` argument).

Alternatively, the function can automatically calculate the values of a chosen effect size or outcome measure (and the corresponding sampling variances) when supplied with the necessary data. The `escalc` function describes which effect size or outcome measures are currently implemented and what data/arguments should then be specified/used. The `measure` argument should then be set to the desired effect size or outcome measure.

Specifying the Model

The function can be used to fit fixed- and random/mixed-effects models, as well as meta-regression models including moderators (the difference between the various models is described in detail in the introductory **metafor-package** help file).

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

One or more moderators can be included in these models via the `mods` argument. A single moderator can be given as a (row or column) vector of length k specifying the values of the moderator. Multiple moderators are specified by giving an appropriate design matrix with k rows and p' columns (e.g., using `mods = cbind(mod1, mod2, mod3)`, where `mod1`, `mod2`, and `mod3` correspond to the names of the variables for the three moderator variables). The intercept is included in the model by default unless `intercept=FALSE`.

Alternatively, one can use standard `formula` syntax to specify the model. In this case, the `mods` argument should be set equal to a one-sided formula of the form `mods = ~ model` (e.g., `mods = ~ mod1 + mod2 + mod3`). Interactions, polynomial terms, and factors can be easily added to the model in this manner. When specifying a model formula via the `mods` argument, the `intercept` argument is ignored. Instead, the inclusion/exclusion of the intercept term is controlled by the specified formula (e.g., `mods = ~ mod1 + mod2 + mod3 - 1` would lead to the removal of the intercept term).

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.

When the observed outcomes and corresponding sampling variances are supplied via the `yi` and `vi` arguments (or `sei` or `weights`), one can also directly specify moderators via the `yi` argument (e.g., `rma(yi ~ mod1 + mod2 + mod3, vi)`). In that case, the `mods` argument is ignored and the inclusion/exclusion of the intercept term again is controlled by the specified formula.

Omnibus Test of Parameters

For models including moderators, an omnibus test of all the model coefficients is conducted that excludes the intercept (the first coefficient) if it is included in the model. If no intercept is included in the model, then the omnibus test includes all of the coefficients in the model including the first. Alternatively, one can manually specify the indices of the coefficients to test via the `btt` argument. For example, use `btt=c(3,4)` to only include the third and fourth coefficient from the model in the test (if an intercept is included in the model, then it corresponds to the first coefficient in the model).

Categorical Moderators

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

Knapp & Hartung Adjustment

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

Value

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

`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.
`k`	number of outcomes included in the model fitting.
`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 is an intercept-only model.
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 fit statistics should also be given, use summary.rma (or use the fitstats.rma function to extract them). Full versus reduced model comparisons in terms of fit statistics and likelihoods can be obtained with anova.rma.uni. Permutation tests for the model coefficient(s) can be obtained with permutest.rma.uni.

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

The residuals.rma, rstandard.rma.uni, and rstudent.rma.uni functions extract raw and standardized residuals. Additional case diagnostics (e.g., to determine influential studies) can be obtained with the influence.rma.uni function. For models without moderators, leave-one-out diagnostics can also be obtained with leave1out.rma.uni.

A confidence interval for the amount of (residual) heterogeneity in the random/mixed-effects model can be obtained with confint.rma.uni.

Forest, funnel, radial, and L'abbe plots (the latter two only for models without moderators) can be obtained with forest.rma, funnel.rma, radial.rma, and labbe.rma. The qqnorm.rma.uni function provides normal QQ plots of the standardized residuals. One can also just call plot.rma.uni on the fitted model object to obtain various plots at once.

Tests for funnel plot asymmetry (which may be indicative of publication bias) can be obtained with ranktest.rma and regtest.rma. For models without moderators, the trimfill.rma.uni method can be used to carry out a trim and fill analysis.

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, deviance.rma, AIC.rma, BIC.rma, hatvalues.rma.uni, and weights.rma.uni.

Note

While the HS, HE, DL, and SJ estimators of τ^2 are based on closed-form solutions, the ML, REML, and EB estimators must be obtained numerically. For this, the rma() function makes use of the Fisher scoring algorithm, which is robust to poor starting values and usually converges quickly (Harville, 1977; Jennrich & Sampson, 1976). By default, the starting value is set equal to the value of the Hedges estimator and the algorithm terminates when the change in the estimated value of τ^2 is smaller than 10^{-5} from one iteration to the next. The maximum number of iterations is 100 by default (which should be sufficient in most cases). A different starting value,

threshold, and maximum number of iterations can be specified via the `control` argument by setting `control=list(tau2.init=value, threshold=value, maxiter=value)` when calling the `rma()` function. The step length of the Fisher scoring algorithm can also be manually adjusted by a desired factor with `control=list(stepadj=value)` (values below 1 will reduce the step length). Information on the evolution of the algorithm is obtained by setting `verbose=TRUE` or 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.

When including moderators in the model, it is possible that the design matrix is not of full rank (i.e., there is a linear relationship between the moderator variables included in the model). In that case, the model cannot be fitted and an error will be issued. For example, two moderators that correlated perfectly would cause this problem. 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 arcsine transformed risk difference and Fisher's r-to-z transformed correlation coefficient) are based on variance stabilizing transformations that also help to make the assumption of known sampling variances much more reasonable.
- When fitting a mixed/random-effects model, τ^2 is estimated and then treated as a known constant thereafter. This ignores the uncertainty in the estimate of τ^2. As a consequence, the standard errors of the parameter estimates tend to be too small, yielding test statistics that are too large and confidence intervals that are not wide enough. The Knapp and Hartung (2003) adjustment can be used to counter this problem, yielding test statistics and confidence intervals whose properties are closer to nominal.
- Most effect size measures are not exactly normally distributed as assumed under the various models. However, the normal approximation usually becomes more accurate for most effect size or outcome measures as the within-study sample sizes increase. Therefore, sufficiently

large within-study sample sizes are (usually) needed to be certain that the tests and confidence intervals have nominal levels/coverage. Again, certain outcome measures (e.g., Fisher's r-to-z transformed correlation coefficient) may be preferable from this perspective as well.

Author(s)

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

References

DerSimonian, R., & Laird, N. (1986). Meta-analysis in clinical trials. *Controlled Clinical Trials*, **7**, 177–188.

Harville, D. A. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, **72**, 320–338.

Hedges, L. V. (1983). A random effects model for effect sizes. *Psychological Bulletin*, **93**, 388–395.

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

Hunter, J. E., & Schmidt, F. L. (2004). *Methods of meta-analysis: Correcting error and bias in research findings* (2nd ed.). Thousand Oaks, CA: Sage.

Jennrich, R. I., & Sampson, P. F. (1976). Newton-Raphson and related algorithms for maximum likelihood variance component estimation. *Technometrics*, **18**, 11–17.

Knapp, G., & Hartung, J. (2003). Improved tests for a random effects meta-regression with a single covariate. *Statistics in Medicine*, **22**, 2693–2710.

Morris, C. N. (1983). Parametric empirical Bayes inference: Theory and applications (with discussion). *Journal of the American Statistical Association*, **78**, 47–65.

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

Sidik, K., & Jonkman, J. N. (2005a). A note on variance estimation in random effects meta-regression. *Journal of Biopharmaceutical Statistics*, **15**, 823–838.

Sidik, K., & Jonkman, J. N. (2005b). Simple heterogeneity variance estimation for meta-analysis. *Journal of the Royal Statistical Society, Series C*, **54**, 367–384.

Viechtbauer, W. (2005). Bias and efficiency of meta-analytic variance estimators in the random-effects model. *Journal of Educational and Behavioral Statistics*, **30**, 261–293.

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

See Also

rma.mh, rma.peto, rma.glmm

Examples

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

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

### random-effects model (method="REML" is default, so technically not needed)
rma(yi, vi, data=dat, method="REML")
rma(yi, sei=sqrt(vi), data=dat, method="REML")
rma(yi, weights=1/vi, data=dat, method="REML")

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

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

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

### using a model formula in the yi argument
rma(yi ~ ablat + year, vi, data=dat, method="REML")

### manual dummy coding of the allocation factor
alloc.random     <- ifelse(dat$alloc == "random",     1, 0)
alloc.alternate  <- ifelse(dat$alloc == "alternate",  1, 0)
alloc.systematic <- ifelse(dat$alloc == "systematic", 1, 0)

### test the allocation factor (in the presence of the other moderators)
### note: "alternate" is the reference level of the allocation factor
### note: the intercept is the first coefficient, so btt=c(2,3)
rma(yi, vi, mods = ~ alloc.random + alloc.systematic + year + ablat,
    data=dat, method="REML", btt=c(2,3))

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

### all pairwise differences using the 'multcomp' package (with Holm's method)
res <- rma(yi, vi, mods = ~ factor(alloc) - 1, data=dat, method="REML")
res
## Not run: summary(glht(res, linfct=contrMat(rep(1,3), type=("Tukey"))), test=adjusted("holm"))

### subgrouping versus using a single model with a factor (subgrouping provides
### an estimate of tau^2 within each subgroup, but the number of studies in each
### subgroup get quite small; the model with the allocation factor provides a
### single estimate of tau^2 based on a larger number of studies, but assumes
### that tau^2 is the same within each subgroup)
res.a <- rma(yi, vi, data=dat, subset=(alloc=="alternate"))
res.r <- rma(yi, vi, data=dat, subset=(alloc=="random"))
res.s <- rma(yi, vi, data=dat, subset=(alloc=="systematic"))
res.a
```

```
res.r
res.s
res <- rma(yi, vi, mods = ~ factor(alloc) - 1, data=dat)
res

### demonstrating that Q_E + Q_M = Q_Total for fixed-effects models
### note: this does not work for random/mixed-effects models, since Q_E and
### Q_Total are calculated under the assumption that tau^2 = 0, while the
### calculation of Q_M incorporates the estimate of tau^2
res <- rma(yi, vi, data=dat, method="FE")
res ### this gives Q_Total
res <- rma(yi, vi, mods = ~ ablat + year, data=dat, method="FE")
res ### this gives Q_E and Q_M
res$QE + res$QM

### decomposition of Q_E into subgroup Q-values
res <- rma(yi, vi, mods = ~ factor(alloc), data=dat)
res

res.a <- rma(yi, vi, data=dat, subset=(alloc=="alternate"))
res.r <- rma(yi, vi, data=dat, subset=(alloc=="random"))
res.s <- rma(yi, vi, data=dat, subset=(alloc=="systematic"))

res.a$QE ### Q-value within subgroup
res.r$QE ### Q-value within subgroup
res.s$QE ### Q-value within subgroup

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

`transf` *Transformation Functions*

Description

A set of transformation functions useful for meta-analyses.

Usage

```
transf.rtoz(xi, ...)
transf.ztor(xi, ...)
transf.logit(xi, ...)
transf.ilogit(xi, ...)
transf.arcsin(xi, ...)
transf.iarcsin(xi, ...)
transf.pft(xi, ni, ...)
transf.ipft(xi, ni, ...)
transf.ipft.hm(xi, targs, ...)
transf.isqrt(xi, ...)
```

```
transf.irft(xi, ti, ...)
transf.iirft(xi, ti, ...)
transf.ahw(xi, ...)
transf.iahw(xi, ...)
transf.abt(xi, ...)
transf.iabt(xi, ...)
transf.ztor.int(xi, targs, ...)
transf.exp.int(xi, targs, ...)
transf.ilogit.int(xi, targs, ...)
```

Arguments

xi	vector of values to be transformed.
ni	vector of sample sizes.
ti	vector of person-times at risk.
targs	list with additional arguments for the transformation function. See 'Details'.
...	other arguments.

Details

The following transformation functions are currently implemented:

- transf.rtoz: Fisher's r-to-z transformation.
- transf.ztor: inverse of the Fisher's r-to-z transformation.
- transf.logit: logit (log odds) transformation.
- transf.ilogit: inverse of the logit transformation.
- transf.arcsin: arcsine transformation.
- transf.iarcsin: inverse of the arcsine transformation.
- transf.pft: Freeman-Tukey (double arcsine) transformation for proportions. See Freeman & Tukey (1950). The xi argument is used to specify the proportions and the ni argument the corresponding sample sizes.
- transf.ipft: inverse of the Freeman-Tukey (double arcsine) transformation for proportions. See Miller (1978).
- transf.ipft.hm: inverse of the Freeman-Tukey (double arcsine) transformation for proportions using the harmonic mean of the sample sizes for the back-transformation. See Miller (1978). The sample sizes are specified via the targs argument (the list element should be called ni).
- transf.isqrt: inverse of the square-root transformation (i.e., function to square a number).
- transf.irft: Freeman-Tukey transformation for incidence rates. See Freeman & Tukey (1950). The xi argument is used to specify the incidence rates and the ti argument the corresponding person-time at risk.
- transf.iirft: inverse of the Freeman-Tukey transformation for incidence rates.
- transf.ahw: Transformation of coefficient alpha as suggested by Hakstian & Whalen (1976).

- `transf.iahw`: Inverse of the transformation of coefficient alpha as suggested by Hakstian & Whalen (1976).
- `transf.abt`: Transformation of coefficient alpha as suggested by Bonett (2002).
- `transf.iabt`: Inverse of the transformation of coefficient alpha as suggested by Bonett (2002).
- `transf.ztor.int`: integral transformation method for the z-to-r transformation.
- `transf.exp.int`: integral transformation method for the exponential transformation.
- `transf.ilogit.int`: integral transformation method for the inverse of the logit transformation.

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

Value

A vector with the transformed values.

Author(s)

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

References

Bonett, D. G. (2002). Sample size requirements for testing and estimating coefficient alpha. *Journal of Educational and Behavioral Statistics*, **27**, 335–340.

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.

Hakstian, A. R., & Whalen, T. E. (1976). A k-sample significance test for independent alpha coefficients. *Psychometrika*, **41**, 219–231.

Miller, J. J. (1978). The inverse of the Freeman-Tukey double arcsine transformation. *American Statistician*, **32**, 138.

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

Examples

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

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

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

### average relative risk with 95% CI using the integral transformation
predict(res, transf=transf.exp.int, targs=list(tau2=res$tau2, lower=-4, upper=4))
```

`trimfill` *Trim and Fill Method*

Description

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

Usage

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

Arguments

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

Details

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

Author(s)

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

References

Duval, S. J., & Tweedie, R. L. (2000a). Trim and fill: A simple funnel-plot-based method of testing and adjusting for publication bias in meta-analysis. *Biometrics*, **56**, 455–463.

Duval, S. J., & Tweedie, R. L. (2000b). A nonparametric "trim and fill" method of accounting for publication bias in meta-analysis. *Journal of the American Statistical Association*, **95**, 89–98.

Duval, S. J. (2005). The trim and fill method. In H. R. Rothstein, A. J. Sutton, & M. Borenstein (Eds.) *Publication bias in meta-analysis: Prevention, assessment, and adjustments* (pp. 127–144). Chichester, England: Wiley.

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

See Also

trimfill.rma.uni

---

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

Description

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

Usage

```
## S3 method for class 'rma.uni'
trimfill(x, side, estimator="L0", maxiter=50, verbose=FALSE, ...)
```

Arguments

x	an object of class "rma.uni".
side	either "left" or "right", indicating on which side of the funnel plot the missing studies should be imputed. If left undefined, the side is chosen within the function depending on the results of Egger's regression test (see regtest for details on this test).
estimator	either "L0" or "R0" indicating the estimator to use for estimating the number of missing studies (default is "L0").
maxiter	integer indicating the maximum number of iterations to use for the trim and fill method (default is 50).
verbose	logical indicating whether information about the evolution of the algorithm should be printed (default is FALSE).
...	other arguments.

Details

The trim and fill method is a nonparametric (rank-based) data augmentation technique proposed by Duval and Tweedie (2000a, 2000b; see also Duval, 2005). The method can be used to estimate the number of studies missing from a meta-analysis due to the suppression of the most extreme results on one side of the funnel plot. The method then augments the observed data so that the funnel plot is more symmetric. The trim and fill method can only be used in the context of the fixed- or random-effects model (i.e., in models without moderators). The method should not be regarded as a way of yielding a more "valid" estimate of the overall effect or outcome, but as a way of examining the sensitivity of the results to one particular selection mechanism (i.e., one particular form of publication bias).

Value

An object of class `c("rma.uni.trimfill","rma.uni","rma")`. The object is a list containing the same components as objects created by `rma.uni`, except that the data are augmented by the trim and fill method. The following components are also added:

`k0`	estimated number of missing studies.
`side`	either `"left"` or `"right"`, indicating on which side of the funnel plot the missing studies (if any) were imputed.
`fill`	a vector of dummy codes, indicating which of the data are the observed (0) and the augmented (1) data.

The results of the fitted model after the data augmentation are printed with the `print.rma.uni` function. Calling `funnel.rma` on the object provides a funnel plot of the observed and augmented data.

Author(s)

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

References

Duval, S. J., & Tweedie, R. L. (2000a). Trim and fill: A simple funnel-plot-based method of testing and adjusting for publication bias in meta-analysis. *Biometrics*, **56**, 455–463.

Duval, S. J., & Tweedie, R. L. (2000b). A nonparametric "trim and fill" method of accounting for publication bias in meta-analysis. *Journal of the American Statistical Association*, **95**, 89–98.

Duval, S. J. (2005). The trim and fill method. In H. R. Rothstein, A. J. Sutton, & M. Borenstein (Eds.) *Publication bias in meta-analysis: Prevention, assessment, and adjustments* (pp. 127–144). Chichester, England: Wiley.

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

See Also

`funnel.rma`

Examples

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

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

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

---

vcov.rma *Variance-Covariance Matrix of Parameter Estimates for rma Objects*

Description

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

Usage

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

Arguments

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

Value

A matrix corresponding to the variance-covariance matrix.

Author(s)

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

References

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

See Also

rma.uni, rma.mh, rma.peto, rma.glmm

Examples

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

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

`weights.rma.uni` *Weights for rma Objects*

Description

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

Usage

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

Arguments

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

Value

A vector with the weights.

Author(s)

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

References

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

See Also

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

Examples

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

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

Index