Package 'meta'

January 3, 2019

Title General Package for Meta-Analysis

Version 4.9-4

Depends R (>= 2.9.1)

Imports grid

Suggests metafor (>= 1.9-9), lme4, numDeriv, BiasedUrn

Date 2019-01-02

URL https://github.com/guido-s/meta http://meta-analysis-with-r.org

Description User-friendly general package providing standard methods for meta-analysis and supporting Schwarzer, Carpenter, and Rücker <DOI:10.1007/978-3-319-21416-0>, ``Meta-Analysis with R'' (2015):
- fixed effect and random effects meta-analysis;
- several plots (forest, funnel, Galbraith / radial, L'Abbe, Baujat, bubble);
- statistical tests and trim-and-fill method to evaluate bias in meta-analysis;
- import data from 'RevMan 5';
- prediction interval, Hartung-Knapp and Paule-Mandel method for random effects model;
- cumulative meta-analysis and leave-one-out meta-analysis;
- meta-regression (if R package 'metafor' is installed);
- generalised linear mixed models (if R packages 'metafor', 'lme4', 'numDeriv', and 'BiasedUrn' are installed);
- produce forest plot summarising several (subgroup) meta-analyses.

License GPL (>= 2)

Encoding UTF-8

NeedsCompilation no

Author Guido Schwarzer [cre, aut]

Maintainer Guido Schwarzer <sc@imbi.uni-freiburg.de>

Repository CRAN

Date/Publication 2019-01-03 09:20:03 UTC

R topics documented:

meta-package *meta: Brief overview of methods and general hints*

Description

R package **meta** is a user-friendly general package providing standard methods for meta-analysis and supporting Schwarzer et al. (2015), http://meta-analysis-with-r.org/.

Details

R package **meta** (Schwarzer, 2007) provides the following statistical methods for meta-analysis.

1. Fixed effect and random effects model:
   - Meta-analysis of continuous outcome data (metacont)
   - Meta-analysis of binary outcome data (metabin)
   - Meta-analysis of incidence rates (metainc)
   - Generic inverse variance meta-analysis (metagen)
   - Meta-analysis of single correlations (metacor)
   - Meta-analysis of single means (metamean)
   - Meta-analysis of single proportions (metaprop)
   - Meta-analysis of single incidence rates (metarate)
2. Several plots for meta-analysis:
   - Forest plot (forest)
   - Funnel plot (funnel)
   - Galbraith plot / radial plot (radial)
   - L'Abbe plot for meta-analysis with binary outcome data (labbe)
   - Baujat plot to explore heterogeneity in meta-analysis (baujat)
   - Bubble plot to display the result of a meta-regression (bubble)
3. Statistical tests for funnel plot asymmetry (metabias) and trim-and-fill method (trimfill) to evaluate bias in meta-analysis
4. Import data from 'RevMan 5' (read.rm5); see also metacr to conduct meta-analysis for a single comparison and outcome from a Cochrane review
5. Prediction interval, Hartung-Knapp and Paule-Mandel method for random effects model (see arguments prediction, hakn, and method.tau, respectively, in meta-analysis functions listed under *1. Fixed effect and random effects model*)
6. Cumulative meta-analysis (metacum) and leave-one-out meta-analysis (metainf)
7. Meta-regression (metareg); if R package **metafor** is installed
8. Generalised linear mixed models (metabin, metainc, metaprop, and metarate); if R packages **metafor**, **lme4**, **numDeriv**, and **BiasedUrn** are installed

The following more advanced statistical methods are provided by add-on R packages:

- Frequentist methods for network meta-analysis (R package **netmeta**)

- Advanced methods to model and adjust for bias in meta-analysis (R package **metasens**)

Results of several meta-analyses can be combined with `metabind`. This is, for example, useful to generate a forest plot with results of subgroup analyses.

See `settings.meta` to learn how to print and specify default meta-analysis methods used during your R session. For example, the function can be used to specify general settings:

- `settings.meta("revman5")`
- `settings.meta("jama")`

The first command can be used to reproduce meta-analyses from Cochrane reviews conducted with *Review Manager 5* (RevMan 5, http://community.cochrane.org/tools/review-production-tools/revman-5) and specifies to use a RevMan 5 layout in forest plots. The second command can be used to generate forest plots following instructions for authors of the *Journal of the American Medical Association* (http://jamanetwork.com/journals/jama/pages/instructions-for-authors).

In addition, `settings.meta` can be used to change individual settings. For example, the following R command specifies the use of the Hartung-Knapp and Paule-Mandel methods, and the printing of prediction intervals in the current R session for any meta-analysis generated after execution of this command:

- `settings.meta(hakn=TRUE, method.tau="PM", prediction=TRUE)`

Type `help(package = "meta")` for a listing of R functions and datasets available in **meta**.

Schwarzer (2007) is the preferred citation in publications for **meta**. Type `citation("meta")` for a BibTeX entry of this publication.

To report problems and bugs

- type `bug.report(package = "meta")` if you do not use RStudio,
- send an email to Guido Schwarzer <sc@imbi.uni-freiburg.de> if you use RStudio.

The development version of **meta** is available on GitHub https://github.com/guido-s/meta.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Schwarzer G (2007), meta: An R package for meta-analysis. *R News*, **7**(3), 40–5. https://cran.r-project.org/doc/Rnews/Rnews_2007-3.pdf

Schwarzer G, Carpenter JR and Rücker G (2015), *Meta-Analysis with R (Use-R!)*. Springer International Publishing, Switzerland. http://www.springer.com/gp/book/9783319214153

amlodipine *Amlodipine for Work Capacity*

Description

Meta-analysis on the effect of amlodipine on work capacity.

This meta-analysis is used as a data example in Hartung and Knapp (2001).

Usage

```
data(amlodipine)
```

Format

A data frame with the following columns:

study Study label

n.amlo Number of observations in amlodipine group

mean.amlo Estimated mean in amlodipine group

var.amlo Variance in amlodipine group

n.plac Number of observations in placebo group

mean.plac Estimated mean in placebo group

var.plac Variance in placebo group

Source

Hartung J & Knapp G (2001), On tests of the overall treatment effect in meta-analysis with normally distributed responses. *Statistics in Medicine*, **20**, 1771–82. doi: 10.1002/sim.791 .

See Also

metacont

Examples

```
data(amlodipine)
#
m <- metacont(n.amlo, mean.amlo, sqrt(var.amlo),
              n.plac, mean.plac, sqrt(var.plac),
              data=amlodipine, studlab=study)

m.hakn <- metacont(n.amlo, mean.amlo, sqrt(var.amlo),
                   n.plac, mean.plac, sqrt(var.plac),
                   data=amlodipine, studlab=study,
                   hakn=TRUE)

# Results for mean difference - see Table III in Hartung and Knapp (2001)
```

```
#
res.md <- rbind(data.frame(summary(m)$fixed)[c("TE", "lower", "upper")],
                data.frame(summary(m)$random)[c("TE", "lower", "upper")],
                data.frame(summary(m.hakn)$random)[c("TE", "lower", "upper")])
#
res.md <- round(res.md, 5)
#
row.names(res.md) <- c("FE", "RE", "RE (HaKn)")
names(res.md) <- c("Absolute difference", "CI lower", "CI upper")
#
res.md
```

`as.data.frame.meta` *Additional functions for objects of class meta*

Description

The `as.data.frame` method returns a data frame containing information on individual studies, e.g., estimated treatment effect and its standard error.

Usage

```
## S3 method for class 'meta'
as.data.frame(x, row.names=NULL, optional=FALSE, ...)
```

Arguments

`x`	An object of class `meta`.
`row.names`	`NULL` or a character vector giving the row names for the data frame.
`optional`	logical. If `TRUE`, setting row names and converting column names (to syntactic names) is optional.
`...`	other arguments

Value

A data frame is returned by the function `as.data.frame`.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

`metabin`, `metacont`, `metagen`, `forest.meta`

Examples

```
data(Fleiss93cont)
#
# Generate additional variable with grouping information
#
Fleiss93cont$group <- c(1,2,1,1,2)
#
# Do meta-analysis without grouping information
#
meta1 <- metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c, study,
                  data=Fleiss93cont, sm="SMD")
#
# Update meta-analysis object and do subgroup analyses
#
summary(update(meta1, byvar=group))

#
# Same result using metacont function directly
#
meta2 <- metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c, study,
                  data=Fleiss93cont, sm="SMD", byvar=group)
summary(meta2)

#
# Compare printout of the following two commands
#
as.data.frame(meta1)
meta1$data
```

---

baujat.meta *Baujat plot to explore heterogeneity in meta-analysis*

Description

Draw a Baujat plot to explore heterogeneity in meta-analysis.

Usage

```
baujat(x, ...)

## S3 method for class 'meta'
baujat(x,
       yscale=1, xlim, ylim,
       xlab="Contribution to overall heterogeneity",
       ylab="Influence on overall result",
       pch=21, cex=1, col="black", bg="darkgray",
       studlab=TRUE, cex.studlab=0.8,
       xmin=0, ymin=0, pos=2, offset=0.5,
       grid=TRUE, col.grid="lightgray", lty.grid="dotted", lwd.grid=par("lwd"),
       pty="s", ...)
```

Arguments

x	An object of class meta.
yscale	Scaling factor for values on y-axis.
xlim	The x limits (min,max) of the plot.
ylim	The y limits (min,max) of the plot.
xlab	A label for the x-axis.
ylab	A label for the y-axis.
pch	The plotting symbol used for individual studies.
cex	The magnification to be used for plotting symbol.
col	A vector with colour of plotting symbols.
bg	A vector with background colour of plotting symbols (only used if pch in 21:25).
studlab	A logical indicating whether study labels should be printed in the graph. A vector with study labels can also be provided (must be of same length as x$TE then).
cex.studlab	The magnification for study labels.
xmin	A numeric specifying minimal value to print study labels (on x-axis).
ymin	A numeric specifying minimal value to print study labels (on y-axis).
pos	A position specifier for study labels (see text).
offset	Offset for study labels (see text).
grid	A logical indicating whether a grid is printed in the plot.
col.grid	Colour for grid lines.
lty.grid	The line type for grid lines.
lwd.grid	The line width for grid lines.
pty	A character specifying type of plot region (see par).
...	Graphical arguments as in par may also be passed as arguments.

Details

Baujat et al. (2002) introduced a scatter plot to explore heterogeneity in meta-analysis. On the x-axis the contribution of each study to the overall heterogeneity statistic (see list object Q of the meta-analysis object x) is plotted. On the y-axis the standardised difference of the overall treatment effect with and without each study is plotted; this quantity describes the influence of each study on the overal treatment effect.

Internally, the metainf function is used to calculate the values on the y-axis.

Value

A data.frame with the following variables:

x	Coordinate on x-axis (contribution to heterogeneity statistic).
y	Coordinate on y-axis (influence on overall treatment effect).

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Baujat B, Mahé C, Pignon JP, Hill C (2002), A graphical method for exploring heterogeneity in meta-analyses: Application to a meta-analysis of 65 trials. *Statistics in Medicine*, **30**, 2641–2652.

See Also

metagen, metainf

Examples

```
data(Olkin95)

m1 <- metabin(event.e, n.e, event.c, n.c, data=Olkin95,
              studlab=author, sm="OR", method="I")

# Generate Baujat plot
baujat(m1)

# Do not print study labels if the x-value is smaller than 4 and the
# y-value is smaller than 1.
baujat(m1, yscale=10, xmin=4, ymin=1)

# Change position of study labels
baujat(m1, yscale=10, xmin=4, ymin=1,
       pos=1, xlim=c(0, 6.5))

# Generate Baujat plot and assign x- and y- coordinates to R object b1
b1 <- baujat(m1)

# Calculate overall heterogeneity statistic
sum(b1$x)
m1$Q
```

---

bubble.metareg *Bubble plot to display the result of a meta-regression*

Description

Draw a bubble plot to display the result of a meta-regression.

Usage

```
bubble(x, ...)

## S3 method for class 'metareg'
bubble(x,
       xlim, ylim, xlab, ylab,
       cex, min.cex=0.5, max.cex=5,
       pch=21, col="black", bg="darkgray",
       lty=1, lwd=1, col.line="black",
       studlab=FALSE, cex.studlab=0.8,
       pos=2, offset=0.5,
       regline=TRUE,
       axes=TRUE, box=TRUE,
       ...)
```

Arguments

x	An object of class metareg.
xlim	The x limits (min,max) of the plot.
ylim	The y limits (min,max) of the plot.
xlab	A label for the x-axis.
ylab	A label for the y-axis.
cex	The magnification to be used for plotting symbols.
min.cex	Minimal magnification for plotting symbols.
max.cex	Maximal magnification for plotting symbols.
pch	The plotting symbol used for individual studies.
col	A vector with colour of plotting symbols.
bg	A vector with background colour of plotting symbols (only used if pch in 21:25).
lty	The line type for the meta-regression line.
lwd	The line width for the meta-regression line.
col.line	Colour for the meta-regression line.
studlab	A logical indicating whether study labels should be printed in the graph. A vector with study labels can also be provided (must be of same length as the numer of studies in the meta-analysis then).
cex.studlab	The magnification for study labels.
pos	A position specifier for study labels (see text).
offset	Offset for study labels (see text).
regline	A logical indicating whether a regression line should be added to the bubble plot.
axes	A logical indicating whether axes should be printed.
box	A logical indicating whether a box should be printed.
...	Graphical arguments as in par may also be passed as arguments.

Details

A bubble plot can be used to display the result of a meta-regression. It is a scatter plot with the treatment effect for each study on the y-axis and the covariate used in the meta-regression on the x-axis. Typically, the size of the plotting symbol is inversely proportional to the variance of the estimated treatment effect (Thompson & Higgins, 2002).

Argument `cex` specifies the plotting size for each individual study. If this argument is missing the weights from the meta-regression model will be used (which typically is a random effects model). Use `weight="fixed"` in order to utilise weights from a fixed effect model to define the size of the plotted symbols (even for a random effects meta-regression). If a vector with individual study weights is provided, the length of this vector must be of the same length as the number of studies.

Arguments `min.cex` and `max.cex` can be used to define the size of the smallest and largest plotting symbol. The plotting size of the most precise study is set to `max.cex` whereas the plotting size of all studies with a plotting size smaller than `min.cex` will be set to `min.cex`.

For a meta-regression with more than one covariate. Only a scatter plot of the first covariate in the regression model is shown. In this case the effect of the first covariate adjusted for other covariates in the meta-regression model is shown.

For a factor or categorial covariate separate bubble plots for each group compared to the baseline group are plotted.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Thompson SG, Higgins JP (2002), How should meta-regression analyses be undertaken and interpreted? *Statistics in Medicine*, **21**, 1559–1573.

See Also

metagen, metainf

Examples

```
data(Fleiss93cont)

# Add some (fictitious) grouping variables:
Fleiss93cont$age <- c(55, 65, 52, 65, 58)
Fleiss93cont$region <- c("Europe", "Europe", "Asia", "Asia", "Europe")

meta1 <- metacont(n.e, mean.e, sd.e,
                  n.c, mean.c, sd.c,
                  data=Fleiss93cont, sm="MD")

mr1 <- metareg(meta1, region)
mr1

bubble(mr1)
bubble(mr1, lwd=2, col.line="blue")
```

```
mr2 <- metareg(meta1, age)
mr2

bubble(mr2, lwd=2, col.line="blue", xlim=c(50, 70))
bubble(mr2, lwd=2, col.line="blue", xlim=c(50, 70), cex="fixed")

# Do not print regression line
#
bubble(mr2, lwd=2, col.line="blue", xlim=c(50, 70), regline=FALSE)
```

`ci` *Calculation of confidence intervals (based on normal approximation or t-distribution)*

Description

Calculation of confidence intervals; based on normal approximation or t-distribution.

Usage

```
ci(TE, seTE, level=0.95, df=NULL, null.effect = 0)
```

Arguments

`TE`	Estimated treatment effect.
`seTE`	Standard error of treatment estimate.
`level`	The confidence level required.
`df`	Degrees of freedom (for confidence intervals based on t-distribution).
`null.effect`	A numeric value specifying the effect under the null hypothesis.

Value

List with components

`TE`	Estimated treatment effect.
`seTE`	Standard error of treatment estimate.
`lower`	Lower confidence limits.
`upper`	Upper confidence limits.
`z`	Test statistic (either z-score or t-score).
`p`	P-value of test with null hypothesis `TE=0`.
`level`	The confidence level required.
`df`	Degrees of freedom (t-distribution).

Note

This function is primarily called from other functions of the library `meta`, e.g. `forest.meta`, `summary.meta`.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

Examples

```
data.frame(ci(170, 10))
data.frame(ci(170, 10, 0.99))
data.frame(ci(1.959964, 1))
data.frame(ci(2.2621571628, 1, df=9))
```

---

cisapride *Cisapride in Non-Ulcer Dispepsia*

Description

Meta-analysis on cisapride in non-ulcer dispepsia.

This meta-analysis is used as a data example in Hartung and Knapp (2001).

Usage

```
data(cisapride)
```

Format

A data frame with the following columns:

study Study label

event.cisa Number of events in cisapride group

n.cisa Number of observations in cisapride group

event.plac Number of events in placebo group

n.plac Number of observations in placebo group

Source

Hartung J & Knapp G (2001), A Refined Method for the Meta-analysis of Controlled Clinical Trials with Binary Outcome. *Statistics in Medicine*, **20**, 3875–89.

See Also

metabin

Examples

```
data(cisapride)

m.or <- metabin(event.cisa, n.cisa, event.plac, n.plac,
                data=cisapride, sm="OR", method="Inverse",
                studlab=study, addincr=TRUE)

m.rr <- metabin(event.cisa, n.cisa, event.plac, n.plac,
                data=cisapride, sm="RR", method="Inverse",
                studlab=study, addincr=TRUE)

m.or.hakn <- metabin(event.cisa, n.cisa, event.plac, n.plac,
                     data=cisapride, sm="OR", method="Inverse",
                     studlab=study, addincr=TRUE,
                     hakn=TRUE)

m.rr.hakn <- metabin(event.cisa, n.cisa, event.plac, n.plac,
                     data=cisapride, sm="RR", method="Inverse",
                     studlab=study, addincr=TRUE,
                     hakn=TRUE)

# Results for log risk ratio - see Table VII in Hartung and Knapp (2001)
#
res.rr <- rbind(data.frame(summary(m.rr)$fixed)[c("TE", "lower", "upper")],
                data.frame(summary(m.rr)$random)[c("TE", "lower", "upper")],
                data.frame(summary(m.rr.hakn)$random)[c("TE", "lower", "upper")])
#
row.names(res.rr) <- c("FE", "RE", "RE (HaKn)")
names(res.rr) <- c("Log risk ratio", "CI lower", "CI upper")
#
res.rr


# Results for log odds ratio (Table VII in Hartung and Knapp 2001)
#
res.or <- rbind(data.frame(summary(m.or)$fixed)[c("TE", "lower", "upper")],
                data.frame(summary(m.or)$random)[c("TE", "lower", "upper")],
                data.frame(summary(m.or.hakn)$random)[c("TE", "lower", "upper")])
#
row.names(res.or) <- c("FE", "RE", "RE (HaKn)")
names(res.or) <- c("Log odds ratio", "CI lower", "CI upper")
#
res.or
```

`Fleiss93` *Aspirin after Myocardial Infarction*

Description

Meta-analysis on aspirin in preventing death after myocardial infarction.

Data example in Fleiss (1993) for meta-analysis with binary outcomes.

Usage

```
data(Fleiss93)
```

Format

A data frame with the following columns:

study Study label

year Year of publication

event.e Number of deaths in aspirin group

n.e Number of observations in aspirin group

event.c Number of deaths in placebo group

n.c Number of observations in placebo group

Source

Fleiss JL (1993), The statistical basis of meta-analysis. *Statistical Methods in Medical Research*, **2**, 121–145.

Examples

```
data(Fleiss93)
metabin(event.e, n.e, event.c, n.c,
        data=Fleiss93,
        studlab=paste(study, year),
        sm="OR", comb.random=FALSE)
```

Fleiss93cont *Mental Health Treatment*

Description

Meta-analysis on the Effect of Mental Health Treatment on Medical Utilisation.

Data example in Fleiss (1993) for meta-analysis with continuous outcomes.

Usage

```
data(Fleiss93cont)
```

Format

A data frame with the following columns:

study Study label

year Year of publication

n.e Number of observations in psychotherapy group

mean.e Estimated mean in psychotherapy group

sd.e Standard deviation in psychotherapy group

n.c Number of observations in control group

mean.c Estimated mean in control group

sd.c Standard deviation in control group

Source

Fleiss JL (1993), The statistical basis of meta-analysis. *Statistical Methods in Medical Research*, **2**, 121–145.

See Also

Fleiss93

Examples

```
data(Fleiss93cont)
metacont(n.e, mean.e, sd.e,
         n.c, mean.c, sd.c,
         data=Fleiss93cont,
         studlab=paste(study, year),
         comb.random=FALSE)
```

---

forest *Forest plot to display the result of a meta-analysis*

Description

Draws a forest plot in the active graphics window (using grid graphics system).

Usage

```
forest(x, ...)

## S3 method for class 'meta'
forest(x, sortvar, studlab=TRUE,
       layout=gs("layout"),
       comb.fixed=x$comb.fixed, comb.random=x$comb.random,
       overall=TRUE,
```

```
text.fixed=NULL,
text.random=NULL,
lty.fixed=2, lty.random=3, col.fixed="black", col.random="black",
prediction=x$prediction,
text.predict=NULL,
subgroup=TRUE,
print.subgroup.labels=TRUE,
bylab=x$bylab, print.byvar=x$print.byvar,
byseparator=x$byseparator,
text.fixed.w=text.fixed, text.random.w=text.random, bysort=FALSE,
pooled.totals=comb.fixed|comb.random, pooled.events=FALSE,
pooled.times=FALSE, study.results=TRUE,
xlab="", xlab.pos,
smlab=NULL, smlab.pos, xlim="symmetric",
allstudies=TRUE,
weight.study, weight.subgroup,
pscale=x$pscale, irscale=x$irscale, irunit=x$irunit,
ref=ifelse(backtransf & is.relative.effect(x$sm), 1, 0),
leftcols=NULL, rightcols=NULL,
leftlabs=NULL, rightlabs=NULL,
lab.e=x$label.e, lab.c=x$label.c,
lab.e.attach.to.col=NULL, lab.c.attach.to.col=NULL,
label.right=x$label.right, label.left=x$label.left, bottom.lr=TRUE,
lab.NA=".",
lab.NA.effect="",
lwd=1,
at=NULL, label=TRUE,
type.study="square", type.fixed="diamond", type.random=type.fixed,
type.subgroup=ifelse(study.results, "diamond", "square"),
col.study="black",
col.square="gray", col.square.lines=col.square,
col.inside="white",
col.diamond="gray",
col.diamond.fixed=col.diamond, col.diamond.random=col.diamond,
col.diamond.lines="black",
col.diamond.lines.fixed=col.diamond.lines,
col.diamond.lines.random=col.diamond.lines,
col.inside.fixed=col.inside,
col.inside.random=col.inside,
col.predict="red", col.predict.lines="black",
col.by="darkgray",
col.label.right="black", col.label.left="black",
hetstat = print.I2 | print.tau2 | print.Q | print.pval.Q | print.Rb,
overall.hetstat = overall & hetstat,
hetlab = "Heterogeneity: ",
resid.hetstat = overall & hetstat,
resid.hetlab = "Residual heterogeneity: ",
print.I2 = comb.fixed | comb.random,
```

```
print.I2.ci = FALSE,
print.tau2 = comb.fixed | comb.random,
print.Q = FALSE,
print.pval.Q = comb.fixed | comb.random,
print.Rb = FALSE,
print.Rb.ci = FALSE,
text.subgroup.nohet = "not applicable",
##
test.overall=gs("test.overall"),
test.overall.fixed=comb.fixed&overall&test.overall,
test.overall.random=comb.random&overall&test.overall,
label.test.overall.fixed, label.test.overall.random,
print.zval=TRUE,
##
test.subgroup,
test.subgroup.fixed, test.subgroup.random,
print.Q.subgroup=TRUE,
label.test.subgroup.fixed, label.test.subgroup.random,
##
test.effect.subgroup,
test.effect.subgroup.fixed, test.effect.subgroup.random,
label.test.effect.subgroup.fixed,
label.test.effect.subgroup.random,
##
fontsize=12,
fs.heading = fontsize,
fs.fixed, fs.random, fs.predict,
fs.fixed.labels, fs.random.labels, fs.predict.labels,
fs.study = fontsize, fs.study.labels = fs.study,
fs.hetstat, fs.test.overall,
fs.test.subgroup, fs.test.effect.subgroup,
fs.axis = fontsize, fs.smlab = fontsize, fs.xlab = fontsize,
fs.lr = fontsize,
ff.heading = "bold",
ff.fixed, ff.random, ff.predict,
ff.fixed.labels, ff.random.labels, ff.predict.labels,
ff.study = "plain", ff.study.labels = ff.study,
ff.hetstat, ff.test.overall,
ff.test.subgroup, ff.test.effect.subgroup,
ff.axis = "plain", ff.smlab = "bold", ff.xlab = "plain",
ff.lr = "plain",
#
squaresize=0.8 / spacing,
#
plotwidth = if (layout != "JAMA") "6cm" else "8cm",
colgap = "2mm",
colgap.left = colgap, colgap.right = colgap,
colgap.studlab = colgap.left, colgap.forest = colgap,
```

```
       colgap.forest.left = colgap.forest,
       colgap.forest.right = colgap.forest,
       #
       calcwidth.pooled=TRUE,
       calcwidth.fixed=calcwidth.pooled,
       calcwidth.random=calcwidth.pooled,
       calcwidth.predict=FALSE,
       calcwidth.hetstat=FALSE, calcwidth.tests=FALSE,
       calcwidth.subgroup=FALSE,
       #
       just=if (layout != "JAMA") "right" else "left",
       just.studlab="left", just.addcols="center",
       just.addcols.left=just.addcols, just.addcols.right=just.addcols,
       #
       spacing = 1,
       addrow, addrow.overall, addrow.subgroups,
       #
       new=TRUE,
       #
       backtransf=x$backtransf,
       digits=gs("digits.forest"), digits.se=gs("digits.se"),
       digits.zval=gs("digits.zval"),
       digits.pval=max(gs("digits.pval")-2, 2),
       digits.pval.Q=max(gs("digits.pval.Q")-2, 2),
       digits.Q=gs("digits.Q"),
       digits.tau2=gs("digits.tau2"),
       digits.I2=max(gs("digits.I2")-1, 0),
       digits.weight=gs("digits.weight"),
       #
       digits.mean = digits, digits.sd = digits.se,
       digits.cor = digits, digits.time = digits,
       #
       scientific.pval = gs("scientific.pval"), big.mark = gs("big.mark"),
       #
       col.i=col.study, weight=weight.study,
       ...)

## S3 method for class 'metabind'
forest(x,
       leftcols, leftlabs,
       rightcols=c("effect", "ci"), rightlabs,
       #
       overall=FALSE, subgroup=FALSE, overall.hetstat=FALSE,
       #
       lab.NA="",
       #
       digits=gs("digits.forest"),
       digits.se=gs("digits.se"),
```

```
digits.zval=gs("digits.zval"),
digits.pval=max(gs("digits.pval") - 2, 2),
digits.pval.Q=max(gs("digits.pval.Q") - 2, 2),
digits.Q=gs("digits.Q"),
digits.tau2=gs("digits.tau2"),
digits.I2=max(gs("digits.I2") - 1, 0),
#
scientific.pval=gs("scientific.pval"),
big.mark=gs("big.mark"),
#
smlab,
...)
```

Arguments

x	An object of class meta or metabind.
sortvar	An optional vector used to sort the individual studies (must be of same length as x$TE).
studlab	A logical indicating whether study labels should be printed in the graph. A vector with study labels can also be provided (must be of same length as x$TE then).
layout	A character string specifying the layout of the forest plot (see Details).
comb.fixed	A logical indicating whether fixed effect estimate should be plotted.
comb.random	A logical indicating whether random effects estimate should be plotted.
overall	A logical indicating whether overall summaries should be plotted. This argument is useful in a meta-analysis with subgroups if summaries should only be plotted on group level.
text.fixed	A character string used in the plot to label the pooled fixed effect estimate.
text.random	A character string used in the plot to label the pooled random effects estimate.
lty.fixed	Line type (pooled fixed effect estimate).
lty.random	Line type (pooled random effects estimate).
col.fixed	Line colour (pooled fixed effect estimate).
col.random	Line colour (pooled random effects estimate).
prediction	A logical indicating whether a prediction interval should be printed.
text.predict	A character string used in the plot to label the prediction interval.
subgroup	A logical indicating whether subgroup results should be shown in forest plot. This argument is useful in a meta-analysis with subgroups if summaries should not be plotted on group level.
print.subgroup.labels	A logical indicating whether subgroup label should be printed.
bylab	A character string with a label for the grouping variable.
print.byvar	A logical indicating whether the name of the grouping variable should be printed in front of the group labels.

`byseparator` A character string defining the separator between label and levels of grouping variable.

`text.fixed.w` A character string to label the pooled fixed effect estimate within subgroups, or a character vector of same length as number of subgroups with corresponging labels.

`text.random.w` A character string to label the pooled random effect estimate within subgroups, or a character vector of same length as number of subgroups with corresponging labels.

`bysort` A logical indicating whether groups should be ordered alphabetically.

`pooled.totals` A logical indicating whether total number of observations should be given in the figure.

`pooled.events` A logical indicating whether total number of events should be given in the figure.

`pooled.times` A logical indicating whether total person time at risk should be given in the figure.

`study.results` A logical indicating whether results for individual studies should be shown in the figure (useful to only plot subgroup results).

`xlab` A label for the x-axis.

`xlab.pos` A numeric specifying the center of the label on the x-axis.

`smlab` A label for the summary measurex (printed at top of figure).

`smlab.pos` A numeric specifying the center of the label for the summary measure.

`xlim` The x limits (min,max) of the plot, or the character "s" to produce symmetric forest plots.

`allstudies` A logical indicating whether studies with inestimable treatment effects should be plotted.

`weight.study` A character string indicating weighting used to determine size of squares or diamonds (argument `type.study`) to plot individual study results. One of missing, `"same"`, `"fixed"`, or `"random"`, can be abbreviated. Plot symbols have the same size for all studies or represent study weights from fixed effect or random effects model.

`weight.subgroup` A character string indicating weighting used to determine size of squares or diamonds (argument `type.subgroup`) to plot subgroup results. One of missing, `"same"`, or `"weight"`, can be abbreviated. Plot symbols have the same size for all subgroup results or represent subgroup weights from fixed effect or random effects model.

`pscale` A numeric giving scaling factor for printing of single event probabilities or risk differences, i.e. if argument `sm` is equal to `"PLOGIT"`, `"PLN"`, `"PRAW"`, `"PAS"`, `"PFT"`, or `"RD"`.

`irscale` A numeric defining a scaling factor for printing of single incidence rates or incidence rate differences, i.e. if argument `sm` is equal to `"IR"`, `"IRLN"`, `"IRS"`, `"IRFT"`, or `"IRD"`.

`irunit` A character specifying the time unit used to calculate rates, e.g. person-years.

`ref` A numerical giving the reference value to be plotted as a line in the forest plot. No reference line is plotted if argument `ref` is equal to `NA`.

`leftcols` A character vector specifying (additional) columns to be plotted on the left side of the forest plot or a logical value (see Details).

`rightcols` A character vector specifying (additional) columns to be plotted on the right side of the forest plot or a logical value (see Details).

`leftlabs` A character vector specifying labels for (additional) columns on left side of the forest plot (see Details).

`rightlabs` A character vector specifying labels for (additional) columns on right side of the forest plot (see Details).

`lab.e` Label to be used for experimental group in table heading.

`lab.c` Label to be used for control group in table heading.

`lab.e.attach.to.col`
A character specifying the column name where label `lab.e` should be attached to in table heading.

`lab.c.attach.to.col`
A character specifying the column name where label `lab.c` should be attached to in table heading.

`label.left` Graph label on left side of forest plot.

`label.right` Graph label on right side of forest plot.

`bottom.lr` A logical indicating whether labels on right and left side should be printed at bottom or top of forest plot.

`lab.NA` A character string to label missing values.

`lab.NA.effect` A character string to label missing values in individual treatment estimates and confidence intervals.

`lwd` The line width, see `par`.

`at` The points at which tick-marks are to be drawn, see `grid.xaxis`.

`label` A logical value indicating whether to draw the labels on the tick marks, or an expression or character vector which specify the labels to use. See `grid.xaxis`.

`type.study` A character string or vector specifying how to plot treatment effects and confidence intervals for individual studies (see Details).

`type.fixed` A character string specifying how to plot treatment effect and confidence interval for fixed effect meta-analysis (see Details).

`type.random` A character string specifying how to plot treatment effect and confidence interval for random effects meta-analysis (see Details).

`type.subgroup` A character string specifying how to plot treatment effect and confidence interval for subgroup results (see Details).

`col.study` The colour for individual study results and confidence limits.

`col.inside` The colour for individual study results and confidence limits if confidence limits are completely within squares.

`col.square` The colour for squares reflecting study's weight in the meta-analysis.

`col.square.lines`
: The colour for the outer lines of squares reflecting study's weight in the meta-analysis.

`col.diamond`
: The colour of diamonds representing the results for fixed effect and random effects models.

`col.diamond.fixed`
: The colour of diamonds for fixed effect estimates.

`col.diamond.random`
: The colour of diamonds for random effects estimates.

`col.diamond.lines`
: The colour of the outer lines of diamonds representing the results for fixed effect and random effects models.

`col.diamond.lines.fixed`
: The colour of the outer lines of diamond for fixed effect estimate.

`col.diamond.lines.random`
: The colour of the outer lines of diamond for random effects estimate.

`col.inside.fixed`
: The colour for result of fixed effect meta-analysis if confidence limit lies completely within square.

`col.inside.random`
: The colour for result of random effects meta-analysis if confidence limit lies completely within square.

`col.predict`
: Background colour of prediction interval.

`col.predict.lines`
: Colour of outer lines of prediction interval.

`col.by`
: The colour to print information on subgroups.

`col.label.right`
: The colour for label on right side of null effect.

`col.label.left`
: The colour for label on left side of null effect.

`hetstat`
: A logical value indicating whether to print results for heterogeneity measures at all.

`overall.hetstat`
: A logical value indicating whether to print heterogeneity measures for overall treatment comparisons. This argument is useful in a meta-analysis with subgroups if heterogeneity statistics should only be printed on subgroup level.

`hetlab`
: Label printed in front of results for heterogeneity measures.

`resid.hetstat`
: A logical value indicating whether to print measures of residual heterogeneity in a meta-analysis with subgroups.

`resid.hetlab`
: Label printed in front of results for residual heterogeneity measures.

`print.I2`
: A logical value indicating whether to print the value of the I-squared statistic.

`print.I2.ci`
: A logical value indicating whether to print the confidence interval of the I-squared statistic.

`print.tau2`
: A logical value indicating whether to print the value of the between-study variance tau-squared.

`print.Q` A logical value indicating whether to print the value of the heterogeneity statistic Q.

`print.pval.Q` A logical value indicating whether to print the p-value of the heterogeneity statistic Q.

`print.Rb` A logical value indicating whether to print the value of the I-squared statistic.

`print.Rb.ci` A logical value indicating whether to print the confidence interval of the I-squared statistic.

`text.subgroup.nohet`
A logical value or character string which is printed to indicate subgroups with less than two studies contributing to meta-analysis (and thus without heterogeneity). If FALSE, heterogeneity statistics are printed (with NAs).

`test.overall` A logical value indicating whether to print results of test for overall effect.

`test.overall.fixed`
A logical value indicating whether to print results of test for overall effect (based on fixed effect model).

`test.overall.random`
A logical value indicating whether to print results of test for overall effect (based on random effects model).

`label.test.overall.fixed`
Label printed in front of results of test for overall effect (based on fixed effect model).

`label.test.overall.random`
Label printed in front of results of test for overall effect (based on random effects model).

`print.zval` A logical value indicating whether z-value for test of treatment effect should be printed.

`test.subgroup` A logical value indicating whether to print results of test for subgroup differences.

`test.subgroup.fixed`
A logical value indicating whether to print results of test for subgroup differences (based on fixed effect model).

`test.subgroup.random`
A logical value indicating whether to print results of test for subgroup differences (based on random effects model).

`print.Q.subgroup`
A logical value indicating whether to print the value of the heterogeneity statistic Q (test for subgroup differences).

`label.test.subgroup.fixed`
Label printed in front of results of test for subgroup differences (based on fixed effect model).

`label.test.subgroup.random`
Label printed in front of results of test for subgroup differences (based on random effects model).

`test.effect.subgroup`
A logical value indicating whether to print results of test for effect in subgroups.

`test.effect.subgroup.fixed`
: A logical value indicating whether to print results of test for effect in subgroups (based on fixed effect model).

`test.effect.subgroup.random`
: A logical value indicating whether to print results of test for effect in subgroups (based on random effects model).

`label.test.effect.subgroup.fixed`
: Label printed in front of results of test for effect in subgroups (based on fixed effect model).

`label.test.effect.subgroup.random`
: Label printed in front of results of test for effect in subgroups (based on random effects model).

`fontsize` : The size of text (in points), see gpar.

`fs.heading` : The size of text for column headings, see gpar.

`fs.fixed` : The size of text for results of fixed effect model, see gpar.

`fs.random` : The size of text for results of random effects model, see gpar.

`fs.predict` : The size of text for results of prediction interval, see gpar.

`fs.fixed.labels`
: The size of text for label of fixed effect model, see gpar.

`fs.random.labels`
: The size of text for label of random effects model, see gpar.

`fs.predict.labels`
: The size of text for label of prediction interval, see gpar.

`fs.study` : The size of text for results of individual studies, see gpar.

`fs.study.labels`
: The size of text for labels of individual studies, see gpar.

`fs.hetstat` : The size of text for heterogeneity measures, see gpar.

`fs.test.overall`
: The size of text of test for overall effect, see gpar.

`fs.test.subgroup`
: The size of text of test of subgroup differences, see gpar.

`fs.test.effect.subgroup`
: The size of text of test of effect in subgroups, see gpar.

`fs.axis` : The size of text on x-axis, see gpar.

`fs.smlab` : The size of text of label for summary measure, see gpar.

`fs.xlab` : The size of text of label on x-axis, see gpar.

`fs.lr` : The size of text of label on left and right side of forest plot, see gpar.

`ff.heading` : The fontface for column headings, see gpar.

`ff.fixed` : The fontface of text for results of fixed effect model, see gpar.

`ff.random` : The fontface of text for results of random effects model, see gpar.

`ff.predict` : The fontface of text for results of prediction interval, see gpar.

`ff.fixed.labels`
: The fontface of text for label of fixed effect model, see `gpar`.

`ff.random.labels`
: The fontface of text for label of random effects model, see `gpar`.

`ff.predict.labels`
: The fontface of text for label of prediction interval, see `gpar`.

`ff.study`
: The fontface of text for results of individual studies, see `gpar`.

`ff.study.labels`
: The fontface of text for labels of individual studies, see `gpar`.

`ff.hetstat`
: The fontface of text for heterogeneity measures, see `gpar`.

`ff.test.overall`
: The fontface of text of test for overall effect, see `gpar`.

`ff.test.subgroup`
: The fontface of text for test of subgroup differences, see `gpar`.

`ff.test.effect.subgroup`
: The fontface of text for test of effect in subgroups, see `gpar`.

`ff.axis`
: The fontface of text on x-axis, see `gpar`.

`ff.smlab`
: The fontface of text of label for summary measure, see `gpar`.

`ff.xlab`
: The fontface of text of label on x-axis, see `gpar`.

`ff.lr`
: The fontface of text of label on left and right side of forest plot, see `gpar`.

`squaresize`
: A numeric used to increase or decrease the size of squares in the forest plot.

`plotwidth`
: Either a character string, e.g., "8cm", "60mm", or "3inch", or a `unit` object specifying width of the forest plot.

`colgap`
: Either a character string or a `unit` object specifying gap between columns printed on left and right side of forest plot.

`colgap.left`
: Either a character string or a `unit` object specifying gap between columns printed on left side of forest plot.

`colgap.right`
: Either a character string or a `unit` object specifying gap between columns printed on right side of forest plot.

`colgap.studlab`
: Either a character string or a `unit` object specifying gap between column with study labels and subsequent column.

`colgap.forest`
: Either a character string or a `unit` object specifying gap between column adjacent to forest plot and the forest plot.

`colgap.forest.left`
: Either a character string or a `unit` object specifying gap between column on the left side of forest plot and the forest plot.

`colgap.forest.right`
: Either a character string or a `unit` object specifying gap between column on the right side of forest plot and the forest plot.

`calcwidth.pooled`
: A logical indicating whether text for fixed effect and random effects model should be considered to calculate width of the column with study labels.

`calcwidth.fixed`
: A logical indicating whether text given in arguments `text.fixed` and `text.fixed.w` should be considered to calculate width of the column with study labels.

`calcwidth.random`
: A logical indicating whether text given in arguments `text.random` and `text.random.w` should be considered to calculate width of the column with study labels.

`calcwidth.predict`
: A logical indicating whether text given in argument `text.predict` should be considered to calculate width of the column with study labels.

`calcwidth.hetstat`
: A logical indicating whether text for heterogeneity statistics should be considered to calculate width of the column with study labels.

`calcwidth.tests`
: A logical indicating whether text for tests of overall effect or subgroup differences should be considered to calculate width of the column with study labels.

`calcwidth.subgroup`
: A logical indicating whether text with subgroup labels should be considered to calculate width of the column with study labels.

`just`
: Justification of text in all columns but columns with study labels and additional variables (possible values: "left", "right", "center").

`just.studlab`
: Justification of text for study labels (possible values: "left", "right", "center").

`just.addcols`
: Justification of text for additional columns (possible values: "left", "right", "center").

`just.addcols.left`
: Justification of text for additional columns on left side of forest plot (possible values: "left", "right", "center"). Can be of same length as number of additional columns on left side of forest plot.

`just.addcols.right`
: Justification of text for additional columns on right side of forest plot (possible values: "left", "right", "center"). Can be of same length as number of additional columns on right side of forest plot.

`spacing`
: A numeric determining line spacing in a forest plot.

`addrow`
: A logical value indicating whether an empty row is printed above and below study results.

`addrow.overall`
: A logical value indicating whether an empty row is printed above overall meta-analysis results.

`addrow.subgroups`
: A logical value indicating whether an empty row is printed between results for subgroups.

`new`
: A logical value indicating whether a new figure should be printed in an existing graphics window.

`backtransf`
: A logical indicating whether results should be back transformed in forest plots. If `backtransf=TRUE`, results for `sm="OR"` are presented as odds ratios rather than log odds ratios and results for `sm="ZCOR"` are presented as correlations rather than Fisher's z transformed correlations, for example.

digits	Minimal number of significant digits for treatment effects, see print.default.
digits.se	Minimal number of significant digits for standard errors, see print.default.
digits.zval	Minimal number of significant digits for z- or t-statistic for test of overall effect, see print.default.
digits.tau2	Minimal number of significant digits for between-study variance, see print.default.
digits.pval	Minimal number of significant digits for p-value of overall treatment effect, see print.default.
digits.pval.Q	Minimal number of significant digits for p-value of heterogeneity test, see print.default.
digits.Q	Minimal number of significant digits for heterogeneity statistic Q, see print.default.
digits.I2	Minimal number of significant digits for I-squared statistic, see print.default.
digits.weight	Minimal number of significant digits for weights, see print.default. digits.cor=NULL, digits.time=NULL,
digits.mean	Minimal number of significant digits for means; only applies to metacont objects.
digits.sd	Minimal number of significant digits for standard deviations; only applies to metacont objects.
digits.cor	Minimal number of significant digits for correlations; only applies to metacor objects.
digits.time	Minimal number of significant digits for times; only applies to metainc and metarate objects.
scientific.pval	A logical specifying whether p-values should be printed in scientific notation, e.g., 1.2345e-01 instead of 0.12345.
big.mark	A character used as thousands separator.
col.i	Deprecated argument (replaced by col.study).
weight	Deprecated argument (replaced by weight.study).
...	Additional graphical arguments (ignored at the moment).

Details

A forest plot, also called confidence interval plot, is drawn in the active graphics window. The forest function is based on the grid graphics system. In order to print the forest plot, (i) resize the graphics window, (ii) either use dev.copy2eps or dev.copy2pdf.

By default, treatment estimates and confidence intervals are plotted in the following way:

- For an individual study, a square with treatment estimate in the center and confidence interval as line extending either side of the square (type.study="square")
- For meta-analysis results, a diamond with treatment estimate in the center and right and left side corresponding to lower and upper confidence limits (type.fixed="diamond", type.random="diamond", and type.subgroup="diamond")

In a forest plot, size of the squares typically reflects the precision of individual treatment estimates based either on the fixed effect (`weight.study="fixed"`) or random effects meta-analysis (`weight.study="random"`). Information from meta-analysis object `x` is utilised if argument `weight.study` is missing. Weights from the fixed effect model are used if argument `x$comb.fixed` is `TRUE`; weights from the random effects model are used if argument `x$comb.random` is `TRUE` and `x$comb.fixed` is `FALSE`. The same square sizes are used if `weight.study="same"`.

Arguments `text.fixed`, `text.random`, and `text.predict` can be used to change the label to identify overall results (fixed effect and random effects model as well as prediction interval). By default the following text is printed:

- "Fixed effect model" (argument `text.fixed`)
- "Random effects model" (`text.random`)
- "Prediction interval" (`text.predict`)

If confidence interval levels are different for individual studies, meta-analysis, and prediciton interval (arguments `level`, `level.comb`, `level.predict` in meta-analysis functions, e.g. `metabin`), additional information is printed, e.g. " (99%-CI)" for a 99% confidence interval in the meta-analysis.

The following arguments can be used to print results for various statistical tests:

Argument	Statistical test
`test.overall.fixed`	Test for overall effect (fixed effect model)
`test.overall.random`	Test for overall effect (random effects model)
`test.effect.subgroup.fixed`	Test for effect in subgroup (FE model)
`test.effect.subgroup.random`	Test for effect in subgroup (RE model)
`test.subgroup.fixed`	Test for subgroup differences (FE model)
`test.subgroup.random`	Test for subgroup differences (RE model)

By default, these arguments are `FALSE`. R function `settings.meta` can be used to change this default for the entire R session. For example, use the following command to always print results of tests for an overall effect:

```
settings.meta(test.overall=TRUE)
```

The arguments `leftcols` and `rightcols` can be used to specify columns which are plotted on the left and right side of the forest plot, respectively. If argument `rightcols` is `FALSE`, no columns will be plotted on the right side. By default, i.e. if arguments `leftcols` and `rightcols` are `NULL` and `layout="meta"`, the following ***columns*** will be printed ***on the right side of the forest plot***:

Meta-analysis results	Value of argument rightcols
No summary	`c("effect", "ci")`
Only fixed effect model	`c("effect", "ci", "w.fixed")`
Only random effects model	`c("effect", "ci", "w.random")`
Both models	`c("effect", "ci", "w.fixed", "w.random")`

By default, estimated treatment effect and corresponding confidence interval will be printed. Depending on arguments `comb.fixed` and `comb.random`, weights of the fixed effect and/or random effects model will be given too. For an object of class `metacum` or `metainf` only the estimated

treatment effect with confidence interval are plotted.

Depending on the class of the meta-analysis object (which is defined by the R function used to generate the object) a different set of ***columns*** is printed ***on the left side of the forest plot***:

Function	Value of argument leftcols
metabin	c("studlab", "event.e", "n.e", "event.c", "n.c")
metacont	c("studlab", "n.e", "mean.e", "sd.e", "n.c", "mean.c", "sd.c")
metacor	c("studlab", "n")
metagen	c("studlab", "TE", "seTE")
metainc	c("studlab", "event.e", "time.e", "event.c", "time.c")
metaprop	c("studlab", "event", "n")
metarate	c("studlab", "event", "time")
metacum	"studlab"
metainf	"studlab"

The arguments leftlabs and rightlabs can be used to specify column headings which are plotted on left and right side of the forest plot, respectively. For certain columns predefined labels exist. If the arguments leftlabs and rightlabs are NULL, the following default labels will be used:

Column:	studlab	TE	seTE	n.e	n.c	n
Label:	"Study"	"TE"	"seTE"	"Total"	"Total"	"Total"

Column:	event.e	event.c	event	mean.e	mean.c
Label:	"Events"	"Events"	"Events"	"Mean"	"Mean"

Column:	sd.e	sd.c	time.e	time.c	effect
Label:	"SD"	"SD"	"Time"	"Time"	x$sm

Column:	ci	effect.ci	w.fixed	w.random
Label:	x$level"%-CI"	*effect*+*ci*	"W(fixed)"	"W(random)"

For additional columns, the column name will be used as label. It is possible to only provide labels for new columns (see Examples). Otherwise the length of leftlabs and rightlabs must be the same as the number of printed columns, respectively. The value NA can be used to specify columns which should use default labels (see Examples).

If argument layout="RevMan5" (and arguments leftcols and rightcols are NULL), the layout for forest plots used for Cochrane reviews (which are generated with Review Manager 5, http://community.cochrane.org/tools/review-production-tools/revman-5) is reproduced:

1. All columns are printed on the left side of the forest plot (see arguments leftcols and rightcols)
2. Tests for overall effect and subgroup differences are printed (test.overall, test.effect.subgroup, test.subgroup)
3. Diamonds representing meta-analysis results are printed in black (diamond.fixed, diamond.random)

4. Color of squares depends on the meta-analysis object (`col.square`, `col.square.lines`)
5. Information on effect measure and meta-analysis method is printed above the forest plot (`smlab`)
6. Label "Study or Subgroup" is printed for meta-analysis with subgroups (`leftlabs`)

If argument `layout="JAMA"` (and arguments `leftcols` and `rightcols` are `NULL`), instructions for authors of the *Journal of the American Medical Association*, see http://jamanetwork.com/journals/jama/pages/instructions-for-authors, are taken into account:

1. Graph labels on right and left side are printed in bold font at top of forest plot (see arguments `bottom.lr` and `ff.lr`)
2. Information on effect measure and level of confidence interval is printed at bottom of forest plot (`xlab`)
3. Tests for overall effect are printed (`test.overall`)
4. Diamonds representing meta-analysis results are printed in lightblue (`diamond.fixed`, `diamond.random`)
5. Squares representing individual study results are printed in darkblue (`col.square`, `col.square.lines`)
6. Between-study variance τ^2 is not printed
7. Empty rows are omitted (`addrow`)
8. Label "Source" is printed instead of "Study" (`leftlabs`)

The following changes are conducted if argument `layout="subgroup"` (and arguments `leftcols` and `rightcols` are `NULL`) and a subgroup analysis was conducted:

1. Individual study results are omitted (see argument `study.results`)
2. Total number of observations is not printed (`pooled.totals`)
3. Label "Subgroup" is printed instead of "Study" (`leftlabs`)

If arguments `lab.e` and `lab.c` are `NULL`, "Experimental" and "Control" are used as labels for experimental and control group, respectively.

Argument `pscale` can be used to rescale single proportions or risk differences, e.g. `pscale=1000` means that proportions are expressed as events per 1000 observations. This is useful in situations with (very) low event probabilities.

Argument `irscale` can be used to rescale single rates or rate differences, e.g. `irscale=1000` means that rates are expressed as events per 1000 time units, e.g. person-years. This is useful in situations with (very) low rates. Argument `irunit` can be used to specify the time unit used in individual studies (default: "person-years"). This information is printed in summaries and forest plots if argument `irscale` is not equal to 1.

A prediction interval for treatment effect of a new study (Higgins et al., 2009) is given in the forest plot if arguments `prediction` and `comb.random` are `TRUE`. For graphical presentation of prediction intervals the approach by Guddat et al. (2012) is used.

Note, in R package **meta**, version 3.0-0 the following arguments have been removed from R function forest.meta: byvar, level, level.comb, level.predict. This functionality is now provided by R function `update.meta` (or directly in R functions, e.g., `metabin`, `metacont`, `metagen`, `metacor`, and `metaprop`).

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Guddat C, Grouven U, Bender R, Skipka G (2012), A note on the graphical presentation of prediction intervals in random-effects meta-analyses. *Systematic Reviews*, **1**, 34.

Higgins JPT, Thompson SG, Spiegelhalter DJ (2009), A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society: Series A*, **172**, 137-159.

See Also

metabin, metacont, metagen, metabind, settings.meta

Examples

```
data(Olkin95)
meta1 <- metabin(event.e, n.e, event.c, n.c,
                 data=Olkin95, subset=c(41,47,51,59),
                 sm="RR", method="I",
                 studlab=paste(author, year))


## Not run:
#
# Do standard (symmetric) forest plot
#
forest(meta1)

## End(Not run)


#
# Layout of forest plot similar to Review Mananager 5
# (see http://community.cochrane.org/tools/review-production-tools/revman-5)
#
# Furthermore, add labels on both sides of forest plot and prediction
# interval
#
forest(meta1, layout="RevMan5", comb.fixed=FALSE,
       label.right="Favours control", col.label.right="red",
       label.left="Favours experimental", col.label.left="green",
       prediction=TRUE)


## Not run:
#
# Sort studies by decreasing treatment effect within year subgroups
#
meta2 <- update(meta1, byvar=ifelse(year < 1987,
                                    "Before 1987", "1987 and later"),
```

```
                print.byvar=FALSE)
forest(meta2,
       sortvar=-TE, comb.random=FALSE)


#
# Forest plot specifying argument xlim
#
forest(meta1, xlim=c(0.01, 10))


#
# Print results of test for overall effect
#
forest(meta1, test.overall.fixed=TRUE, test.overall.random=TRUE)


#
# Forest plot with 'classic' layout used in
# R package meta, version < 1.6-0
#
forest(meta1, col.square="black", hetstat=FALSE)


#
# Change set of columns printed on left side
# of forest plot
#
forest(meta1, comb.random=FALSE,
       leftcols="studlab")


#
# Do not print columns on right side of forest plot
#
forest(meta1, rightcols=FALSE)


#
# Change study label to "Author"
#
forest(meta1, comb.random=FALSE,
       leftlabs=c("Author", NA, NA, NA, NA))


#
# Just give effect estimate and 95
# on right side of forest plot (in one column)
#
forest(meta1, rightcols=c("effect.ci"))


#
```

```
# Just give effect estimate and 95
# on right side of forest plot
#
forest(meta1, rightcols=c("effect", "ci"))


#
# 1. Change order of columns on left side
# 2. Attach labels to columns 'event.e' and 'event.c'
#    instead of columns 'n.e' and 'n.c'
#
forest(meta1,
       leftcols=c("studlab", "n.e", "event.e", "n.c", "event.c"),
       lab.e.attach.to.col="event.e",
       lab.c.attach.to.col="event.c")


#
# Specify column labels only for newly created variables
# 'year' and 'author' (which are part of dataset Olkin95)
#
forest(meta1,
       leftcols=c("studlab", "event.e", "n.e", "event.c", "n.c",
                  "author", "year"),
       leftlabs=c("Author", "Year of Publ"))


#
# Center text in all columns
#
forest(meta1,
       leftcols=c("studlab", "event.e", "n.e", "event.c", "n.c",
                  "author", "year"),
       leftlabs=c("Author", "Year of Publ"), hetstat=FALSE,
       just="center", just.addcols="center", just.studlab="center")

#
# Same result
#
forest(meta1,
       leftcols=c("studlab", "event.e", "n.e", "event.c", "n.c",
                  "author", "year"),
       leftlabs=c("Author", "Year of Publ"), hetstat=FALSE,
       just="c", just.addcols="c", just.studlab="c")


#
# Change some fontsizes and fontfaces
#
forest(meta1,
       fs.study=10, ff.study="italic",
       fs.study.label=11, ff.study.label="bold",
       fs.axis=5, ff.axis="italic",
```

```
       ff.smlab="bold.italic",
       ff.fixed="plain", ff.hetstat="plain")


#
# Change some colours
#
forest(meta1,
       col.diamond="green", col.diamond.lines="red",
       col.study=c("green", "blue", "red", "orange"),
       col.square="pink", col.square.lines="black")


#
# Sort by weight in fixed effect model
#
forest(meta1, sortvar=1/w.fixed, comb.random=FALSE)


#
# Sort by decreasing weight in fixed effect model
#
forest(meta1, sortvar=-1/w.fixed, comb.random=FALSE)


#
# Sort by size of treatment effect
#
forest(meta1, sortvar=TE, comb.random=FALSE)


#
# Sort by size of treatment effect
#
forest(meta1, sortvar=-TE, comb.random=FALSE)


#
# Sort by decreasing year of publication
#
forest(meta1, sortvar=-year, comb.random=FALSE)


#
# Print results of test for subgroup differences (random effects model)
#
forest(meta2,
       sortvar=-TE, comb.fixed=FALSE,
       test.subgroup.random=TRUE)

#
# Print only subgroup results
#
```

```
forest(meta2, layout="subgroup")

#
# Print only subgroup results
# (and consider text for heterogeneity measures in width of subgroup
#  column)
#
forest(meta2, layout="subgroup", calcwidth.hetstat=TRUE)

## End(Not run)
```

`funnel.meta` *Plot to assess funnel plot asymmetry*

Description

Draw a funnel plot or radial plot (also called Galbraith plot) to assess funnel plot asymmetry in the active graphics window.

A contour-enhanced funnel plot can be produced for assessing causes of funnel plot asymmetry.

Usage

```
funnel(x, ...)

radial(x, ...)

## Default S3 method:
funnel(x, y,
       xlim=NULL, ylim=NULL, xlab=NULL, ylab=NULL,
       comb.fixed=FALSE, comb.random=FALSE,
       axes=TRUE,
       pch=21, text=NULL, cex=1,
       lty.fixed=2, lty.random=9,
       lwd=1, lwd.fixed=lwd, lwd.random=lwd,
       col="black", bg="darkgray",
       col.fixed="black", col.random="black",
       log="", yaxis="se", sm="",
       contour.levels=NULL, col.contour,
       ref=ifelse(backtransf & is.relative.effect(sm), 1, 0),
       level=NULL,
       studlab=FALSE, cex.studlab=0.8, pos.studlab = 2,
       backtransf=TRUE, ...)

## S3 method for class 'meta'
funnel(x,
       xlim=NULL, ylim=NULL, xlab=NULL, ylab=NULL,
       comb.fixed=x$comb.fixed, comb.random=x$comb.random,
```

```
       axes=TRUE,
       pch=if (!inherits(x, "trimfill")) 21 else ifelse(x$trimfill, 1, 21),
       text=NULL, cex=1,
       lty.fixed=2, lty.random=9,
       lwd=1, lwd.fixed=lwd, lwd.random=lwd,
       col="black", bg="darkgray",
       col.fixed="black", col.random="black",
       log="", yaxis="se",
       contour.levels=NULL, col.contour,
       ref=ifelse(backtransf & is.relative.effect(x$sm), 1, 0),
       level=x$level,
       studlab=FALSE, cex.studlab=0.8, pos.studlab = 2,
       ref.triangle = FALSE,
       lty.ref = 1, lwd.ref = lwd, col.ref = "black",
       lty.ref.triangle = 5,
       backtransf=x$backtransf, ...)

## Default S3 method:
radial(x, y, xlim=NULL, ylim=NULL,
       xlab="Inverse of standard error",
       ylab="Standardised treatment effect (z-score)",
       comb.fixed=TRUE, axes=TRUE,
       pch=1, text=NULL, cex=1, col=NULL,
       level=NULL, ...)

## S3 method for class 'meta'
radial(x, xlim=NULL, ylim=NULL,
       xlab="Inverse of standard error",
       ylab="Standardised treatment effect (z-score)",
       comb.fixed=TRUE, axes=TRUE,
       pch=1, text=NULL, cex=1, col=NULL,
       level=NULL, ...)
```

Arguments

x	An object of class meta, or estimated treatment effect in individual studies.
y	Standard error of estimated treatment effect.
xlim	The x limits (min,max) of the plot.
ylim	The y limits (min,max) of the plot.
xlab	A label for the x-axis.
ylab	A label for the y-axis.
comb.fixed	A logical indicating whether the pooled fixed effect estimate should be plotted.
comb.random	A logical indicating whether the pooled random effects estimate should be plotted.
axes	A logical indicating whether axes should be drawn on the plot.
pch	The plotting symbol used for individual studies.

text	A character vector specifying the text to be used instead of plotting symbol.
cex	The magnification to be used for plotting symbol.
lty.fixed	Line type (pooled fixed effect estimate).
lty.random	Line type (pooled random effects estimate).
col	A vector with colour of plotting symbols.
bg	A vector with background colour of plotting symbols (only used if pch in 21:25).
col.fixed	Color of line representing fixed effect estimate.
col.random	Color of line representing random effects estimate.
lwd	The line width for confidence intervals (if level is not NULL).
lwd.fixed	The line width for fixed effect estimate (if comb.fixed is not NULL).
lwd.random	The line width for random effects estimate (if comb.random is not NULL).
log	A character string which contains "x" if the x-axis is to be logarithmic, "y" if the y-axis is to be logarithmic and "xy" or "yx" if both axes are to be logarithmic (applies only to function funnel).
yaxis	A character string indicating which type of weights are to be used. Either "se", "invvar", "invse", or "size" (applies only to function funnel).
sm	A character string indicating underlying summary measure, e.g., "RD", "RR", "OR", "ASD", "HR", "MD", "SMD", or "ROM" (applies only to function funnel).
contour.levels	A numeric vector specifying contour levels to produce contour-enhanced funnel plot.
col.contour	Colour of contours.
ref	Reference value (null effect) used to produce contour-enhanced funnel plot.
level	The confidence level utilised in the plot. For the funnel plot, confidence limits are not drawn if yaxis="size".
studlab	A logical indicating whether study labels should be printed in the graph. A vector with study labels can also be provided (must be of same length as x$TE then).
cex.studlab	Size of study labels, see argument cex in text.
pos.studlab	Position of study labels, see argument pos in text.
ref.triangle	A logical indicating whether reference value (null effect) should be printed.
lty.ref	Line type (reference value).
lwd.ref	The line width for the reference value and corresponding confidence intervals (if ref.triangle is TRUE and level is not NULL).
col.ref	Color of line representing reference value.
lty.ref.triangle	Line type (confidence intervals of reference value).
backtransf	A logical indicating whether results for relative summary measures (argument sm equal to "OR", "RR", "HR", or "IRR") should be back transformed in funnel plots. If backtransf=TRUE, results for sm="OR" are printed as odds ratios rather than log odds ratios, for example.
...	Graphical arguments as in par may also be passed as arguments.

Details

A funnel plot or radial plot, also called Galbraith plot, is drawn in the active graphics window. If `comb.fixed` is TRUE, the pooled estimate of the fixed effect model is plotted. If `level` is not NULL, the corresponding confidence limits are drawn.

In the funnel plot, if `yaxis` is `"se"`, the standard error of the treatment estimates is plotted on the y-axis which is likely to be the best choice (Sterne & Egger, 2001). Other possible choices for `yaxis` are `"invvar"` (inverse of the variance), `"invse"` (inverse of the standard error), and `"size"` (study size).

For `yaxis!="size"`, contour-enhanced funnel plots can be produced (Peters et al., 2008) by specifying the contour levels (argument `contour.levels`). By default (argument `col.contour` missing), suitable gray levels will be used to distinguish the contours. Different colours can be chosen by argument `col.contour`.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>, Petra Graham <pgraham@efs.mq.edu.au>

References

Galbraith RF (1988a), Graphical display of estimates having differing standard errors. *Technometrics*, **30**, 271–281.

Galbraith RF (1988b), A note on graphical presentation of estimated odds ratios from several clinical trials. *Statistics in Medicine*, **7**, 889–894.

Light RJ & Pillemer DB (1984), *Summing Up. The Science of Reviewing Research*. Cambridge: Harvard University Press.

Peters JL, Sutton AJ, Jones DR, Abrams KR, 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 JAC & Egger M (2001), Funnel plots for detecting bias in meta-analysis: Guidelines on choice of axis. *Journal of Clinical Epidemiology*, **54**, 1046–1055.

See Also

`metabias`, `metabin`, `metagen`

Examples

```
data(Olkin95)
meta1 <- metabin(event.e, n.e, event.c, n.c,
                 data=Olkin95, subset=c(41,47,51,59),
                 studlab=paste(author, year),
                 sm="RR", method="I")


#
# Radial plot
#
radial(meta1, level=0.95)
```

```
oldpar <- par(mfrow=c(2, 2))

#
# Funnel plots
#
funnel(meta1)
#
# Same result as code above:
#
funnel(meta1$TE, meta1$seTE, sm="RR",
       comb.fixed=TRUE, level=0.95)
#
# Funnel plot with confidence intervals,
# fixed effect estimate and contours
#
cc <- funnel(meta1, comb.fixed=TRUE,
             level=0.95, contour=c(0.9, 0.95, 0.99))$col.contour
legend(0.05, 0.05,
       c("0.1 > p > 0.05", "0.05 > p > 0.01", "< 0.01"), fill=cc)
#
# Contour-enhanced funnel plot with user-chosen colours
#
funnel(meta1, comb.fixed=TRUE,
       level=0.95, contour=c(0.9, 0.95, 0.99),
       col.contour=c("darkgreen", "green", "lightgreen"),
       lwd=2, cex=2, pch=16, studlab=TRUE, cex.studlab=1.25)
legend(0.05, 0.05,
       c("0.1 > p > 0.05", "0.05 > p > 0.01", "< 0.01"),
       fill=c("darkgreen", "green", "lightgreen"))

par(oldpar)
```

---

gs *Get default for a meta-analysis setting.*

Description

Get default for a meta-analysis setting in R package **meta**.

Usage

```
gs(x)
```

Arguments

x	A character string holding a settings name.

Details

This function can be used to get the default for a meta-analysis setting defined using `settings.meta`.

This function is primarily used to define default settings in meta-analysis functions, e.g. `metabin` or `metacont`. A list of all arguments with current settings is printed using the command `settings.meta("print")`.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

`settings.meta`

Examples

```
# Get default setting for Hartung-Knapp method
#
gs("hakn")

# Get default setting for summary measure in metabin()
#
gs("smbin")
```

`labbe.metabin` *L'Abbe plot for meta-analysis with binary outcomes*

Description

Draw a L'Abbé plot for meta-analysis with binary outcomes.

Usage

```
labbe(x, ...)

## Default S3 method:
labbe(x, y,
      xlim, ylim,
      xlab = NULL, ylab = NULL,
      TE.fixed = NULL, TE.random = NULL,
      comb.fixed = !is.null(TE.fixed), comb.random = !is.null(TE.random),
      backtransf = TRUE,
      axes = TRUE,
      pch = 21, text = NULL, cex = 1,
      col = "black", bg = "lightgray",
      lwd = 1, lwd.fixed = lwd, lwd.random = lwd,
      lty.fixed = 2, lty.random = 9,
      col.fixed = col, col.random = col,
```

```
      nulleffect = TRUE,
      lwd.nulleffect = lwd, col.nulleffect = "lightgray",
      sm = "", weight,
      studlab = FALSE, cex.studlab = 0.8,
      label.e = NULL, label.c = NULL,
      ...)

## S3 method for class 'metabin'
labbe(x,
      xlim, ylim,
      xlab = NULL, ylab = NULL,
      TE.fixed = x$TE.fixed,
      TE.random = x$TE.random,
      comb.fixed = x$comb.fixed,
      comb.random = x$comb.random,
      backtransf = x$backtransf,
      axes = TRUE,
      pch = 21, text = NULL, cex = 1,
      col = "black", bg = "lightgray",
      lwd = 1, lwd.fixed = lwd, lwd.random = lwd,
      lty.fixed = 2, lty.random = 9,
      col.fixed = col, col.random = col,
      nulleffect = TRUE,
      lwd.nulleffect = lwd, col.nulleffect = "lightgray",
      sm = x$sm, weight,
      studlab = FALSE, cex.studlab = 0.8,
      label.e = x$label.e, label.c = x$label.c,
      ...)
```

Arguments

x	An object of class metabin. Alternatively, the x coordinates of points of the L'Abbé plot.
y	The y coordinates of the L'Abbé plot, if argument x is not an object of class metabin.
xlim	The x limits (min, max) of the plot.
ylim	The y limits (min, max) of the plot.
xlab	A label for the x-axis.
ylab	A label for the y-axis.
TE.fixed	A numeric or vector specifying combined fixed effect estimate(s).
TE.random	A numeric or vector specifying combined random effects estimate(s).
comb.fixed	A logical indicating whether the pooled fixed effect estimate should be plotted.
comb.random	A logical indicating whether the pooled random effects estimate should be plotted.
backtransf	A logical indicating which values should be printed on x- and y-axis (see Details).

axes	A logical indicating whether axes should be drawn on the plot.
pch	The plotting symbol used for individual studies.
text	A character vector specifying the text to be used instead of plotting symbol.
cex	The magnification to be used for plotting symbol.
col	A vector with colour of plotting symbols.
bg	A vector with background colour of plotting symbols (only used if pch in 21:25).
lwd	The line width.
lwd.fixed	The line width(s) for fixed effect estimate(s) (if comb.fixed is not NULL or FALSE).
lwd.random	The line width(s) for random effects estimate(s) (if comb.random is not NULL or FALSE).
lty.fixed	Line type(s) for fixed effect estimate(s).
lty.random	Line type(s) for random effects estimate(s).
col.fixed	Color of line(s) for fixed effect estimate(s).
col.random	Color of line(s) for random effects estimate(s).
nulleffect	A logical indicating whether line for null effect should be added to the plot..
lwd.nulleffect	Width of line for null effect.
col.nulleffect	Color of line for null effect.
sm	A character string indicating underlying summary measure, i.e., "RD", "RR", "OR", or "ASD".
weight	Either a numeric vector specifying relative sizes of plotting symbols or a character string indicating which type of plotting symbols is to be used for individual treatment estimates. One of missing (see Details), "same", "fixed", or "random", can be abbreviated. Plot symbols have the same size for all studies or represent study weights from fixed effect or random effects model.
studlab	A logical indicating whether study labels should be printed in the graph. A vector with study labels can also be provided (must be of same length as x$event.e then).
cex.studlab	Size of study labels.
label.e	Label for experimental group.
label.c	Label for control group.
...	Graphical arguments as in par may also be passed as arguments.

Details

A L'Abbé plot is a scatter plot with the risk in the control group on the x-axis and the risk in the experimental group on the y-axis (L'Abbé et al., 1987). It can be used to evaluate heterogeneity in meta-analysis. Furthermore, this plot can aid to choose a summary measure (odds ratio, risk ratio, risk difference) that will result in more consistent results.

If argument backtransf is TRUE (default), event probabilities will be printed on x- and y-axis. Otherwise, transformed event probabilities will be printed as defined by the summary measure, i.e.,

log odds of probabilities for odds ratio as summary measure (`sm = "OR"`), log probabilities for `sm = "RR"`, and arcsine-transformed probabilities for `sm = "ASD"`.

If `comb.fixed` is TRUE, the pooled estimate of the fixed effect model is plotted as a line. If `comb.random` is TRUE, the pooled estimate of the random effects model is plotted as a line.

Information from object `x` is utilised if argument `weight` is missing. Weights from the fixed effect model are used (`weight = "fixed"`) if argument `x$comb.fixed` is `TRUE`; weights from the random effects model are used (`weight = "random"`) if argument `x$comb.random` is `TRUE` and `x$comb.fixed` is `FALSE`.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

L'Abbé KA, Detsky AS, O'Rourke K (1987), Meta-analysis in clinical research. *Annals of Internal Medicine*, **107**, 224–233.

See Also

`metabin`

Examples

```
data(Olkin95)
meta1 <- metabin(event.e, n.e, event.c, n.c,
                 data = Olkin95,
                 studlab = paste(author, year),
                 sm = "RR", method = "I")

#
# L'Abbe plot for risk ratio
#
labbe(meta1)

# L'Abbe plot for odds ratio
#
labbe(meta1, sm = "OR")
# same plot
labbe(update(meta1, sm = "OR"))

# L'Abbe plot for risk difference
#
labbe(meta1, sm = "RD")

# L'Abbe plot on log odds scale
#
labbe(meta1, sm = "OR", backtransf = FALSE)

# L'Abbe plot for odds ratio with coloured lines for various treatment
# effects (defined as log odds ratios)
```

```
#
mycols <- c("blue", "yellow", "green", "red",
            "green", "yellow", "blue")
labbe(meta1, sm = "OR",
      comb.random = FALSE,
      TE.fixed = log(c(1 / 10, 1 / 5, 1 / 2,
                       1, 2, 5, 10)),
      col.fixed = mycols, lwd.fixed = 2)

# L'Abbe plot on log odds scale with coloured lines for various
# treatment effects (defined as log odds ratios)
#
labbe(meta1, sm = "OR",
      comb.random = FALSE,
      TE.fixed = log(c(1 / 10, 1 / 5, 1 / 2,
                       1, 2, 5, 10)),
      col.fixed = mycols, lwd.fixed = 2,
      backtransf = FALSE)
```

`metabias` *Test for funnel plot asymmetry*

Description

Test for funnel plot asymmetry, based on rank correlation or linear regression method.

Usage

```
metabias(x, ...)

## S3 method for class 'meta'
metabias(x, method.bias=x$method.bias,
         plotit=FALSE, correct=FALSE, k.min=10, ...)

## Default S3 method:
metabias(x, seTE, method.bias="linreg",
         plotit=FALSE, correct=FALSE, k.min=10, ...)
```

Arguments

`x`	An object of class `meta` or estimated treatment effect in individual studies.
`seTE`	Standard error of estimated treatment effect (mandatory if `x` not of class `meta`).
`method.bias`	A character string indicating which test is to be used. Either `"rank"`, `"linreg"`, `"mm"`, `"count"`, `"score"`, or `"peters"`, can be abbreviated.
`plotit`	A logical indicating whether a plot should be produced for method.bias `"rank"`, `"linreg"`, `"mm"`, or `"score"`.
`correct`	A logical indicating whether a continuity corrected statistic is used for rank correlation methods `"rank"` and `"count"`.

k.min	Minimum number of studies to perform test for funnel plot asymmetry.
...	Additional arguments (ignored at the moment).

Details

Following recommendations by Sterne et al. (2011), by default, a test for funnel plot asymmetry is only conducted if the number of studies is ten or larger (argument k.min=10). This behaviour can be changed by setting a smaller value for argument k.min. Note, the minimum number of studies is three.

If argument method.bias is "rank", the test statistic is based on the rank correlation between standardised treatment estimates and variance estimates of estimated treatment effects; Kendall's tau is used as correlation measure (Begg & Mazumdar, 1994). The test statistic follows a standard normal distribution. By default (if correct is FALSE), no continuity correction is utilised (Kendall & Gibbons, 1990).

If argument method.bias is "linreg", the test statistic is based on a weighted linear regression of the treatment effect on its standard error (Egger et al., 1997). The test statistic follows a t distribution with number of studies - 2 degrees of freedom.

If argument method.bias is "mm", the test statistic is based on a weighted linear regression of the treatment effect on its standard error using the method of moments estimator for the additive between-study variance component (method 3a in Thompson, Sharp, 1999). The test statistic follows a t distribution with number of studies - 2 degrees of freedom.

If argument method.bias is "peters", the test statistic is based on a weighted linear regression of the treatment effect on the inverse of the total sample size using the variance of the average event rate as weights (Peters et al., 2006). The test statistic follows a t distribution with number of studies - 2 degrees of freedom. This test is available for meta-analyses comparing two binary outcomes or combining single proportions, i.e. generated with functions metabin and metaprop.

The following tests for funnel plot asymmetry are only available for meta-analyses comparing two binary outcomes, i.e. meta-analyses generated with the metabin function.

If argument method.bias is "count", the test statistic is based on the rank correlation between a standardised cell frequency and the inverse of the variance of the cell frequency; Kendall's tau is used as correlation measure (Schwarzer et al., 2007). The test statistic follows a standard normal distribution. By default (if correct is FALSE), no continuity correction is utilised (Kendall & Gibbons, 1990).

If argument method.bias is "score", the test statistic is based on a weighted linear regression utilising efficient score and score variance (Harbord et al., 2006). The test statistic follows a t distribution with number of studies - 2 degrees of freedom.

In order to calculate an arcsine test for funnel plot asymmetry (Rücker et al., 2008), one has to use the metabin function with argument sm="ASD" as input to the metabias command. The three arcsine tests described in Rücker et al. (2008) can be calculated by setting method.bias to "rank", "linreg" and "mm", respectively.

If argument method.bias is missing, the Harbord test (method.bias="score") is used for the odds ratio as effect measure and the Egger test (method.bias="linreg") for other effect measures (Sterne et al., 2011).

No test for funnel plot asymmetry is conducted in meta-analyses with subgroups.

Value

A list with class "htest" containing the following components if a test for funnel plot asymmetry is conducted:

estimate	The estimated degree of funnel plot asymmetry, with name "ks" or "bias" corresponding to the method employed, i.e., rank correlation or regression method.
statistic	The value of the test statistic.
parameters	The degrees of freedom of the test statistic in the case that it follows a t distribution.
p.value	The p-value for the test.
alternative	A character string describing the alternative hypothesis.
method	A character string indicating what type of test was used.
data.name	A character string giving the names of the data.
title	Title of Cochrane review.
complab	Comparison label.
outclab	Outcome label.
version	Version of R package **meta** used to create object.

Or a list with the following elements if test is not conducted due to the number of studies:

k	Number of studies in meta-analysis.
k.min	Minimum number of studies to perform test for funnel plot asymmetry.
version	Version of R package **meta** used to create object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

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

Egger M, Smith GD, Schneider M & Minder C (1997), Bias in meta-analysis detected by a simple, graphical test. *British Medical Journal*, **315**, 629–634.

Harbord RM, Egger M & Sterne J (2006), A modified test for small-study effects in meta-analyses of controlled trials with binary endpoints. *Statistics in Medicine*, **25**, 3443–3457.

Kendall M & Gibbons JD (1990), *Rank Correlation Methods*. London: Edward Arnold.

Peters JL, Sutton AJ, Jones DR, Abrams KR & Rushton L (2006), Comparison of two methods to detect publication bias in meta-analysis. *Journal of the American Medical Association*, **295**, 676–680.

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

Schwarzer G, Antes G & Schumacher M (2007), A test for publication bias in meta-analysis with sparse binary data. *Statistics in Medicine*, **26**, 721–733.

Sterne, JAC et al. (2011), Recommendations for Examining and Interpreting Funnel Plot Asymmetry in Meta-Analyses of Randomised Controlled Trials. *BMJ (Clinical research ed.)*, **343**, 1, doi: 10.1136/bmj.d4002 .

Thompson SG & Sharp, SJ (1999), Explaining heterogeneity in meta-analysis: A comparison of methods, *Statistics in Medicine*, **18**, 2693–2708.

See Also

funnel, funnel.meta, metabin, metacont, metagen

Examples

```
data(Olkin95)
meta1 <- metabin(event.e, n.e, event.c, n.c,
                 data=Olkin95, subset=1:10,
                 sm="RR", method="I")

metabias(meta1)
metabias(meta1, plotit=TRUE)

metabias(meta1, method.bias="rank")
metabias(meta1, method.bias="rank", correct=TRUE)


metabias(meta1, method.bias="count")

metabias(meta1, method.bias="linreg")$p.value


#
# Arcsine test (based on linear regression):
#
meta1.as <- metabin(event.e, n.e, event.c, n.c,
                    data=Olkin95, subset=1:10,
                    sm="ASD", method="I")
metabias(meta1.as)
#
# Same result (using function metabias.default):
#
metabias(meta1.as$TE, meta1.as$seTE)


#
# No test for funnel plot asymmetry calculated:
#
meta2 <- metabin(event.e, n.e, event.c, n.c,
                 data=Olkin95, subset=1:5,
                 sm="RR", method="I")
metabias(meta2)

meta3 <- metabin(event.e, n.e, event.c, n.c,
                 data=Olkin95, subset=1:2,
```

```
                  sm="RR", method="I")
metabias(meta3)

# Test for funnel plot asymmetry calculated
# (use of argument k.min):
#
metabias(meta2, k.min=5)
```

---

metabin *Meta-analysis of binary outcome data*

Description

Calculation of fixed effect and random effects estimates (risk ratio, odds ratio, risk difference, or arcsine difference) for meta-analyses with binary outcome data. Mantel-Haenszel, inverse variance, Peto method, and generalised linear mixed model (GLMM) are available for pooling. For GLMMs, the rma.glmm function from R package **metafor** (Viechtbauer 2010) is called internally.

Usage

```
metabin(event.e, n.e, event.c, n.c, studlab,
        data=NULL, subset=NULL, exclude=NULL,
        method=ifelse(tau.common, "Inverse", gs("method")),
        sm=
        ifelse(!is.na(charmatch(tolower(method), c("peto", "glmm"),
                                nomatch = NA)),
                      "OR", gs("smbin")),
        incr=gs("incr"), allincr=gs("allincr"),
        addincr=gs("addincr"), allstudies=gs("allstudies"),
        MH.exact=gs("MH.exact"), RR.cochrane=gs("RR.cochrane"),
        model.glmm = "UM.FS",
        level=gs("level"), level.comb=gs("level.comb"),
        comb.fixed=gs("comb.fixed"), comb.random=gs("comb.random"),
        hakn=gs("hakn"),
        method.tau=
        ifelse(!is.na(charmatch(tolower(method), "glmm", nomatch = NA)),
               "ML", gs("method.tau")),
        tau.preset=NULL, TE.tau=NULL,
        tau.common=gs("tau.common"),
        prediction=gs("prediction"), level.predict=gs("level.predict"),
        method.bias=ifelse(sm=="OR", "score", gs("method.bias")),
        backtransf=gs("backtransf"), pscale = 1,
        title=gs("title"), complab=gs("complab"), outclab="",
        label.e=gs("label.e"), label.c=gs("label.c"),
        label.left=gs("label.left"), label.right=gs("label.right"),
        byvar, bylab, print.byvar=gs("print.byvar"),
        byseparator = gs("byseparator"),
```

```
        print.CMH=gs("print.CMH"),
        keepdata=gs("keepdata"),
        warn=gs("warn"),
        control=NULL,
        ...)
```

Arguments

event.e	Number of events in experimental group.
n.e	Number of observations in experimental group.
event.c	Number of events in control group.
n.c	Number of observations in control group.
studlab	An optional vector with study labels.
data	An optional data frame containing the study information, i.e., event.e, n.e, event.c, and n.c.
subset	An optional vector specifying a subset of studies to be used.
exclude	An optional vector specifying studies to exclude from meta-analysis, however, to include in printouts and forest plots.
method	A character string indicating which method is to be used for pooling of studies. One of "Inverse", "MH", "Peto", or "GLMM", can be abbreviated.
sm	A character string indicating which summary measure ("RR", "OR", "RD", or "ASD") is to be used for pooling of studies, see Details.
incr	Could be either a numerical value which is added to each cell frequency for studies with a zero cell count or the character string "TACC" which stands for treatment arm continuity correction, see Details.
allincr	A logical indicating if incr is added to each cell frequency of all studies if at least one study has a zero cell count. If FALSE (default), incr is added only to each cell frequency of studies with a zero cell count.
addincr	A logical indicating if incr is added to each cell frequency of all studies irrespective of zero cell counts.
allstudies	A logical indicating if studies with zero or all events in both groups are to be included in the meta-analysis (applies only if sm is equal to "RR" or "OR").
MH.exact	A logical indicating if incr is not to be added to all cell frequencies for studies with a zero cell count to calculate the pooled estimate based on the Mantel-Haenszel method.
RR.cochrane	A logical indicating if 2*incr instead of 1*incr is to be added to n.e and n.c in the calculation of the risk ratio (i.e., sm="RR") for studies with a zero cell. This is used in RevMan 5, the Cochrane Collaboration's program for preparing and maintaining Cochrane reviews.
model.glmm	A character string indicating which GLMM should be used. One of "UM.FS", "UM.RS", "CM.EL", and "CM.AL", see Details.
level	The level used to calculate confidence intervals for individual studies.
level.comb	The level used to calculate confidence intervals for pooled estimates.

comb.fixed	A logical indicating whether a fixed effect meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
prediction	A logical indicating whether a prediction interval should be printed.
level.predict	The level used to calculate prediction interval for a new study.
hakn	A logical indicating whether the method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
method.tau	A character string indicating which method is used to estimate the between-study variance τ^2. Either "DL", "PM", "REML", "ML", "HS", "SJ", "HE", or "EB", can be abbreviated.
tau.preset	Prespecified value for the square-root of the between-study variance τ^2.
TE.tau	Overall treatment effect used to estimate the between-study variance τ^2.
tau.common	A logical indicating whether tau-squared should be the same across subgroups.
method.bias	A character string indicating which test for funnel plot asymmetry is to be used. Either "rank", "linreg", "mm", "count", "score", or "peters", can be abbreviated. See function metabias
backtransf	A logical indicating whether results for odds ratio (sm="OR") and risk ratio (sm="RR") should be back transformed in printouts and plots. If TRUE (default), results will be presented as odds ratios and risk ratios; otherwise log odds ratios and log risk ratios will be shown.
pscale	A numeric defining a scaling factor for printing of risk differences.
title	Title of meta-analysis / systematic review.
complab	Comparison label.
outclab	Outcome label.
label.e	Label for experimental group.
label.c	Label for control group.
label.left	Graph label on left side of forest plot.
label.right	Graph label on right side of forest plot.
byvar	An optional vector containing grouping information (must be of same length as event.e).
bylab	A character string with a label for the grouping variable.
print.byvar	A logical indicating whether the name of the grouping variable should be printed in front of the group labels.
byseparator	A character string defining the separator between label and levels of grouping variable.
print.CMH	A logical indicating whether result of the Cochran-Mantel-Haenszel test for overall effect should be printed.
keepdata	A logical indicating whether original data (set) should be kept in meta object.
warn	A logical indicating whether warnings should be printed (e.g., if incr is added to studies with zero cell frequencies).
control	An optional list to control the iterative process to estimate the between-study variance tau^2. This argument is passed on to rma.uni or rma.glmm, respectively.
...	Additional arguments passed on to rma.glmm function.

Details

Treatment estimates and standard errors are calculated for each study. The following measures of treatment effect are available:

- Risk ratio (sm="RR")
- Odds ratio (sm="OR")
- Risk difference (sm="RD")
- Arcsine difference (sm="ASD")

For several arguments defaults settings are utilised (assignments using gs function). These defaults can be changed using the settings.meta function.

Internally, both fixed effect and random effects models are calculated regardless of values chosen for arguments comb.fixed and comb.random. Accordingly, the estimate for the random effects model can be extracted from component TE.random of an object of class "meta" even if argument comb.random=FALSE. However, all functions in R package **meta** will adequately consider the values for comb.fixed and comb.random. E.g. function print.meta will not print results for the random effects model if comb.random=FALSE.

By default, both fixed effect and random effects models are considered (see arguments comb.fixed and comb.random). If method is "MH" (default), the Mantel-Haenszel method is used to calculate the fixed effect estimate; if method is "Inverse", inverse variance weighting is used for pooling; if method is "Peto", the Peto method is used for pooling. For the Peto method, Peto's log odds ratio, i.e. (O - E) / V and its standard error sqrt(1 / V) with O - E and V denoting "Observed minus Expected" and "V", are utilised in the random effects model. Accordingly, results of a random effects model using sm="Peto" can be (slightly) different to results from a random effects model using sm="MH" or sm="Inverse".

A distinctive and frequently overlooked advantage of binary endpoints is that individual patient data (IPD) can be extracted from a two-by-two table. Accordingly, statistical methods for IPD, i.e., logistic regression and generalised linear mixed models, can be utilised in a meta-analysis of binary outcomes (Stijnen et al., 2010; Simmonds et al., 2014). These methods are available (argument method = "GLMM") for the odds ratio as summary measure by calling the rma.glmm function from R package **metafor** internally. Four different GLMMs are available for meta-analysis with binary outcomes using argument model.glmm (which corresponds to argument model in the rma.glmm function):

- Logistic regression model with fixed study effects (default)

  (model.glmm = "UM.FS", i.e., **U**nconditional **M**odel - **F**ixed **S**tudy effects)
- Mixed-effects logistic regression model with random study effects

  (model.glmm = "UM.RS", i.e., **U**nconditional **M**odel - **R**andom **S**tudy effects)
- Generalised linear mixed model (conditional Hypergeometric-Normal)

  (model.glmm = "CM.EL", i.e., **C**onditional **M**odel - **E**xact **L**ikelihood)
- Generalised linear mixed model (conditional Binomial-Normal)

  (model.glmm = "CM.AL", i.e., **C**onditional **M**odel - **A**pproximate **L**ikelihood)

Details on these four GLMMs as well as additional arguments which can be provided using argument '...' in metabin are described in rma.glmm where you can also find information on the

iterative algorithms used for estimation. Note, regardless of which value is used for argument model.glmm, results for two different GLMMs are calculated: fixed effect model (with fixed treatment effect) and random effects model (with random treatment effects).

For studies with a zero cell count, by default, 0.5 is added to all cell frequencies of these studies; if incr is "TACC" a treatment arm continuity correction is used instead (Sweeting et al., 2004; Diamond et al., 2007). For odds ratio and risk ratio, treatment estimates and standard errors are only calculated for studies with zero or all events in both groups if allstudies is TRUE. This continuity correction is used both to calculate individual study results with confidence limits and to conduct meta-analysis based on the inverse variance method. For Peto method and GLMMs no continuity correction is used. For the Mantel-Haenszel method, by default (if MH.exact is FALSE), incr is added to all cell frequencies of a study with a zero cell count in the calculation of the pooled risk ratio or odds ratio as well as the estimation of the variance of the pooled risk difference, risk ratio or odds ratio. This approach is also used in other software, e.g. RevMan 5 and the Stata procedure metan. According to Fleiss (in Cooper & Hedges, 1994), there is no need to add 0.5 to a cell frequency of zero to calculate the Mantel-Haenszel estimate and he advocates the exact method (MH.exact=TRUE). Note, estimates based on exact Mantel-Haenszel method or GLMM are not defined if the number of events is zero in all studies either in the experimental or control group.

Argument byvar can be used to conduct subgroup analysis for all methods but GLMMs. Instead use the metareg function for GLMMs which can also be used for continuous covariates.

A prediction interval for treatment effect of a new study is calculated (Higgins et al., 2009) if arguments prediction and comb.random are TRUE.

R function update.meta can be used to redo the meta-analysis of an existing metabin object by only specifying arguments which should be changed.

For the random effects, the method by Hartung and Knapp (2001) is used to adjust test statistics and confidence intervals if argument hakn=TRUE. For GLMMs, a method similar to Knapp and Hartung (2003) is implemented, see description of argument tdist in rma.glmm.

The DerSimonian-Laird estimate (1986) is used in the random effects model if method.tau="DL". The iterative Paule-Mandel method (1982) to estimate the between-study variance is used if argument method.tau="PM". Internally, R function paulemandel is called which is based on R function mpaule.default from R package **metRology** from S.L.R. Ellison <s.ellison at lgc.co.uk>.

If R package **metafor** (Viechtbauer 2010) is installed, the following methods to estimate the between-study variance τ^2 (argument method.tau) are also available:

- Restricted maximum-likelihood estimator (method.tau="REML")
- Maximum-likelihood estimator (method.tau="ML")
- Hunter-Schmidt estimator (method.tau="HS")
- Sidik-Jonkman estimator (method.tau="SJ")
- Hedges estimator (method.tau="HE")
- Empirical Bayes estimator (method.tau="EB").

For these methods the R function rma.uni of R package **metafor** is called internally. See help page of R function rma.uni for more details on these methods to estimate between-study variance.

Value

An object of class `c("metabin", "meta")` with corresponding `print`, `summary`, and `forest` functions. The object is a list containing the following components:

`event.e, n.e, event.c, n.c, studlab, exclude,`

`sm, method, incr, allincr, addincr,`

`allstudies, MH.exact, RR.cochrane, model.glmm, warn,`

`level, level.comb, comb.fixed, comb.random,`

`hakn, method.tau, tau.preset, TE.tau, method.bias,`

`tau.common, title, complab, outclab,`

`label.e, label.c, label.left, label.right,`

`byvar, bylab, print.byvar, byseparator`
: As defined above.

`TE, seTE`
: Estimated treatment effect and standard error of individual studies.

`lower, upper`
: Lower and upper confidence interval limits for individual studies.

`zval, pval`
: z-value and p-value for test of treatment effect for individual studies.

`w.fixed, w.random`
: Weight of individual studies (in fixed and random effects model).

`TE.fixed, seTE.fixed`
: Estimated overall treatment effect, e.g., log risk ratio or risk difference, and standard error (fixed effect model).

`lower.fixed, upper.fixed`
: Lower and upper confidence interval limits (fixed effect model).

`zval.fixed, pval.fixed`
: z-value and p-value for test of overall treatment effect (fixed effect model).

`TE.random, seTE.random`
: Estimated overall treatment effect, e.g., log risk ratio or risk difference, and standard error (random effects model).

`lower.random, upper.random`
: Lower and upper confidence interval limits (random effects model).

`zval.random, pval.random`
: z-value or t-value and corresponding p-value for test of overall treatment effect (random effects model).

`prediction, level.predict`
: As defined above.

`seTE.predict`
: Standard error utilised for prediction interval.

`lower.predict, upper.predict`
: Lower and upper limits of prediction interval.

k	Number of studies combined in meta-analysis.
Q	Heterogeneity statistic Q.
df.Q	Degrees of freedom for heterogeneity statistic.
pval.Q	P-value of heterogeneity test.
Q.LRT	Heterogeneity statistic for likelihood-ratio test (only if method = "GLMM").
df.Q.LRT	Degrees of freedom for likelihood-ratio test
pval.Q.LRT	P-value of likelihood-ratio test.
tau	Square-root of between-study variance.
se.tau	Standard error of square-root of between-study variance.
C	Scaling factor utilised internally to calculate common tau-squared across subgroups.
Q.CMH	Cochran-Mantel-Haenszel test statistic for overall effect.
df.Q.CMH	Degrees of freedom for Cochran-Mantel-Haenszel test statistic.
pval.Q.CMH	P-value of Cochran-Mantel-Haenszel test.
incr.e, incr.c	Increment added to cells in the experimental and control group, respectively.
sparse	Logical flag indicating if any study included in meta-analysis has any zero cell frequencies.
doublezeros	Logical flag indicating if any study has zero cell frequencies in both treatment groups.
df.hakn	Degrees of freedom for test of treatment effect for Hartung-Knapp method (only if hakn=TRUE).
k.MH	Number of studies combined in meta-analysis using Mantel-Haenszel method.
bylevs	Levels of grouping variable - if byvar is not missing.
TE.fixed.w, seTE.fixed.w	Estimated treatment effect and standard error in subgroups (fixed effect model) - if byvar is not missing.
lower.fixed.w, upper.fixed.w	Lower and upper confidence interval limits in subgroups (fixed effect model) - if byvar is not missing.
zval.fixed.w, pval.fixed.w	z-value and p-value for test of treatment effect in subgroups (fixed effect model) - if byvar is not missing.
TE.random.w, seTE.random.w	Estimated treatment effect and standard error in subgroups (random effects model) - if byvar is not missing.
lower.random.w, upper.random.w	Lower and upper confidence interval limits in subgroups (random effects model) - if byvar is not missing.
zval.random.w, pval.random.w	z-value or t-value and corresponding p-value for test of treatment effect in subgroups (random effects model) - if byvar is not missing.

`w.fixed.w, w.random.w`
: Weight of subgroups (in fixed and random effects model) - if `byvar` is not missing.

`df.hakn.w`
: Degrees of freedom for test of treatment effect for Hartung-Knapp method in subgroups - if `byvar` is not missing and `hakn=TRUE`.

`n.harmonic.mean.w`
: Harmonic mean of number of observations in subgroups (for back transformation of Freeman-Tukey Double arcsine transformation) - if `byvar` is not missing.

`event.e.w`
: Number of events in experimental group in subgroups - if `byvar` is not missing.

`n.e.w`
: Number of observations in experimental group in subgroups - if `byvar` is not missing.

`event.c.w`
: Number of events in control group in subgroups - if `byvar` is not missing.

`n.c.w`
: Number of observations in control group in subgroups - if `byvar` is not missing.

`k.w`
: Number of studies combined within subgroups - if `byvar` is not missing.

`k.all.w`
: Number of all studies in subgroups - if `byvar` is not missing.

`Q.w.fixed`
: Overall within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.w.random`
: Overall within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing (only calculated if argument `tau.common` is TRUE).

`df.Q.w`
: Degrees of freedom for test of overall within subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.w.fixed`
: P-value of within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.w.random`
: P-value of within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`Q.b.fixed`
: Overall between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.b.random`
: Overall between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`df.Q.b`
: Degrees of freedom for test of overall between subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.b.fixed`
: P-value of between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.b.random`
: P-value of between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`tau.w`
: Square-root of between-study variance within subgroups - if `byvar` is not missing.

`C.w`
: Scaling factor utilised internally to calculate common tau-squared across subgroups - if `byvar` is not missing.

`H.w`
: Heterogeneity statistic H within subgroups - if `byvar` is not missing.

`lower.H.w, upper.H.w`
: Lower and upper confidence limti for heterogeneity statistic H within subgroups - if `byvar` is not missing.

`I2.w`
: Heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`lower.I2.w, upper.I2.w`
: Lower and upper confidence limti for heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`keepdata`
: As defined above.

`data`
: Original data (set) used in function call (if `keepdata=TRUE`).

`subset`
: Information on subset of original data used in meta-analysis (if `keepdata=TRUE`).

`.glmm.fixed`
: GLMM object generated by call of `rma.glmm` function (fixed effect model).

`.glmm.random`
: GLMM object generated by call of `rma.glmm` function (random effects model).

`call`
: Function call.

`version`
: Version of R package **meta** used to create object.

`version.metafor`
: Version of R package **metafor** used for GLMMs.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Cooper H & Hedges LV (1994), *The Handbook of Research Synthesis*. Newbury Park, CA: Russell Sage Foundation.

Diamond GA, Bax L, Kaul S (2007), Uncertain Effects of Rosiglitazone on the Risk for Myocardial Infarction and Cardiovascular Death. *Annals of Internal Medicine*, **147**, 578–581.

DerSimonian R & Laird N (1986), Meta-analysis in clinical trials. *Controlled Clinical Trials*, **7**, 177–188.

Fleiss JL (1993), The statistical basis of meta-analysis. *Statistical Methods in Medical Research*, **2**, 121–145.

Greenland S & Robins JM (1985), Estimation of a common effect parameter from sparse follow-up data. *Biometrics*, **41**, 55–68.

Hartung J & Knapp G (2001), A Refined Method for the Meta-analysis of Controlled Clinical Trials with Binary Outcome. *Statistics in Medicine*, **20**, 3875–89.

Higgins JPT, Thompson SG, Spiegelhalter DJ (2009), A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society: Series A*, **172**, 137–159.

Knapp G & Hartung J (2003), Improved Tests for a Random Effects Meta-regression with a Single Covariate. *Statistics in Medicine*, **22**, 2693–710, doi: 10.1002/sim.1482 .

Review Manager (RevMan) [Computer program]. Version 5.3. Copenhagen: The Nordic Cochrane Centre, The Cochrane Collaboration, 2014.

Paule RC & Mandel J (1982), Consensus values and weighting factors. *Journal of Research of the National Bureau of Standards*, **87**, 377–385.

Pettigrew HM, Gart JJ, Thomas DG (1986), The bias and higher cumulants of the logarithm of a binomial variate. *Biometrika*, **73**, 425–435.

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

Simmonds MC, Higgins JP (2014), A general framework for the use of logistic regression models in meta-analysis. *Statistical Methods in Medical Research*.

StataCorp. 2011. *Stata Statistical Software: Release 12*. College Station, TX: StataCorp LP.

Stijnen T, Hamza TH, 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**, 3046–67.

Sweeting MJ, Sutton AJ, Lambert PC (2004), What to add to nothing? Use and avoidance of continuity corrections in meta-analysis of sparse data. *Statistics in Medicine*, **23**, 1351–1375.

Viechtbauer W (2010), Conducting Meta-Analyses in R with the Metafor Package. *Journal of Statistical Software*, **36**, 1–48.

See Also

update.meta, forest, funnel, metabias, metacont, metagen, metareg, print.meta

Examples

```
#
# Calculate odds ratio and confidence interval for a single study
#
metabin(10, 20, 15, 20, sm = "OR")

#
# Different results (due to handling of studies with double zeros)
#
metabin(0, 10, 0, 10, sm = "OR")
metabin(0, 10, 0, 10, sm = "OR", allstudies = TRUE)


#
# Use subset of Olkin (1995) to conduct meta-analysis based on inverse
# variance method (with risk ratio as summary measure)
#
data(Olkin95)
meta1 <- metabin(event.e, n.e, event.c, n.c,
                 data = Olkin95, subset = c(41, 47, 51, 59),
                 method = "Inverse")
summary(meta1)


#
# Use different subset of Olkin (1995)
#
meta2 <- metabin(event.e, n.e, event.c, n.c,
                 data = Olkin95, subset = Olkin95$year < 1970,
```

```
                 method = "Inverse", studlab = author)
summary(meta2)
forest(meta2)
#
# Meta-analysis with odds ratio as summary measure
#
meta3 <- metabin(event.e, n.e, event.c, n.c,
                 data = Olkin95, subset = Olkin95$year < 1970,
                 sm = "OR", method = "Inverse", studlab = author)
# Same meta-analysis result using 'update.meta' function
meta3 <- update(meta2, sm = "OR")
summary(meta3)
#
# Meta-analysis based on Mantel-Haenszel method
# (with odds ratio as summary measure)
#
meta4 <- update(meta3, method = "MH")
summary(meta4)
#
# Meta-analysis based on Peto method
# (only available for odds ratio as summary measure)
#
meta5 <- update(meta3, method = "Peto")
summary(meta5)


## Not run:
#
# Meta-analysis using generalised linear mixed models
# (only if R packages 'metafor' and 'lme4' are available)
#
if (suppressMessages(require(metafor, quietly = TRUE, warn = FALSE)) &
    require(lme4, quietly = TRUE)) {
#
# Logistic regression model with (k = 4) fixed study effects
# (default: model.glmm = "UM.FS")
#
meta6 <- metabin(event.e, n.e, event.c, n.c,
                 data = Olkin95, subset = Olkin95$year < 1970,
                 method = "GLMM")
# Same results:
meta6 <- update(meta2, method = "GLMM")
summary(meta6)
#
# Mixed-effects logistic regression model with random study effects
# (warning message printed due to argument 'nAGQ')
#
meta7 <- update(meta6, model.glmm = "UM.RS")
#
# Use additional argument 'nAGQ' for internal call of 'rma.glmm' function
#
meta7 <- update(meta6, model.glmm = "UM.RS", nAGQ = 1)
summary(meta7)
```

```
#
# Generalised linear mixed model (conditional Hypergeometric-Normal)
# (R package 'BiasedUrn' must be available)
#
if (require(BiasedUrn, quietly = TRUE)) {
 meta8 <- update(meta6, model.glmm = "CM.EL")
 summary(meta8)
}
#
# Generalised linear mixed model (conditional Binomial-Normal)
#
meta9 <- update(meta6, model.glmm = "CM.AL")
summary(meta9)

#
# Logistic regression model with (k = 70) fixed study effects
# (about 18 seconds with Intel Core i7-3667U, 2.0GHz)
#
meta10 <- metabin(event.e, n.e, event.c, n.c,
                  data = Olkin95, method = "GLMM")
summary(meta10)
#
# Mixed-effects logistic regression model with random study effects
# - about 50 seconds with Intel Core i7-3667U, 2.0GHz
# - several warning messages, e.g. "failure to converge, ..."
#
summary(update(meta10, model.glmm = "UM.RS"))
#
# Conditional Hypergeometric-Normal GLMM
# - long computation time (about 12 minutes with Intel Core i7-3667U, 2.0GHz)
# - estimation problems for this very large dataset:
#   * warning that Choleski factorization of Hessian failed
#   * confidence interval for treatment effect smaller in random
#     effects model compared to fixed effect model
#
if (require(BiasedUrn, quietly = TRUE)) {
 system.time(meta11 <- update(meta10, model.glmm = "CM.EL"))
 summary(meta11)
}
#
# Generalised linear mixed model (conditional Binomial-Normal)
# (less than 1 second with Intel Core i7-3667U, 2.0GHz)
#
summary(update(meta10, model.glmm = "CM.AL"))
}

## End(Not run)
```

`metabind` *Combine meta-analysis objects*

Description

This function can be used to combine meta-analysis objects and is, for example, useful to generate a forest plot with results of subgroup analyses.

Usage

```
metabind(..., name, pooled, backtransf, outclab)
```

Arguments

...	Any number of meta-analysis objects (see Details).
name	An optional character vector providing descriptive names for the meta-analysis objects.
pooled	A character string indicating whether results of a fixed effect or random effects model should be considered. Either "fixed" or "random", can be abbreviated.
backtransf	A logical indicating whether results should be back transformed in printouts and plots. If backtransf=TRUE (default), results for sm="OR" are printed as odds ratios rather than log odds ratios, for example.
outclab	Outcome label for all meta-analyis objects.

Details

This function can be used to combine meta-analysis objects and is, for example, useful to generate a forest plot with results of subgroup analyses.

Value

An object of class c("metabind", "meta") with corresponding print, summary, and forest functions. See metagen for more information on list elements.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

metagen, forest.metabind

Examples

```
data(Fleiss93cont)

# Add some (fictitious) grouping variables:
Fleiss93cont$age <- c(55, 65, 55, 65, 55)
Fleiss93cont$region <- c("Europe", "Europe", "Asia", "Asia", "Europe")

meta1 <- metacont(n.e, mean.e, sd.e,
                  n.c, mean.c, sd.c,
                  data = Fleiss93cont, sm = "MD")
```

```
mu1 <- update(meta1, byvar = age, bylab = "Age group")
mu2 <- update(meta1, byvar = region, bylab = "Region")

mb1 <- metabind(mu1, mu2)
mb1

forest(mb1)
```

`metacont` *Meta-analysis of continuous outcome data*

Description

Calculation of fixed and random effects estimates for meta-analyses with continuous outcome data; inverse variance weighting is used for pooling.

Usage

```
metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c, studlab,
         data=NULL, subset=NULL, exclude=NULL,
         sm=gs("smcont"), pooledvar=gs("pooledvar"),
         method.smd=gs("method.smd"), sd.glass=gs("sd.glass"),
         exact.smd=gs("exact.smd"),
         level=gs("level"), level.comb=gs("level.comb"),
         comb.fixed=gs("comb.fixed"), comb.random=gs("comb.random"),
         hakn=gs("hakn"),
         method.tau=gs("method.tau"), tau.preset=NULL, TE.tau=NULL,
         tau.common=gs("tau.common"),
         prediction=gs("prediction"), level.predict=gs("level.predict"),
         method.bias=gs("method.bias"),
         backtransf=gs("backtransf"),
         title=gs("title"), complab=gs("complab"), outclab="",
         label.e=gs("label.e"), label.c=gs("label.c"),
         label.left=gs("label.left"), label.right=gs("label.right"),
         byvar, bylab, print.byvar=gs("print.byvar"),
         byseparator=gs("byseparator"),
         keepdata=gs("keepdata"),
         warn=gs("warn"),
         control=NULL)
```

Arguments

`n.e`	Number of observations in experimental group.
`mean.e`	Estimated mean in experimental group.
`sd.e`	Standard deviation in experimental group.
`n.c`	Number of observations in control group.

mean.c	Estimated mean in control group.
sd.c	Standard deviation in control group.
studlab	An optional vector with study labels.
data	An optional data frame containing the study information.
subset	An optional vector specifying a subset of studies to be used.
exclude	An optional vector specifying studies to exclude from meta-analysis, however, to include in printouts and forest plots.
level	The level used to calculate confidence intervals for individual studies.
level.comb	The level used to calculate confidence intervals for pooled estimates.
comb.fixed	A logical indicating whether a fixed effect meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
prediction	A logical indicating whether a prediction interval should be printed.
level.predict	The level used to calculate prediction interval for a new study.
hakn	A logical indicating whether the method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
method.tau	A character string indicating which method is used to estimate the between-study variance τ^2. Either "DL", "PM", "REML", "ML", "HS", "SJ", "HE", or "EB", can be abbreviated.
tau.preset	Prespecified value for the square-root of the between-study variance τ^2.
TE.tau	Overall treatment effect used to estimate the between-study variance tau-squared.
tau.common	A logical indicating whether tau-squared should be the same across subgroups.
method.bias	A character string indicating which test is to be used. Either "rank", "linreg", or "mm", can be abbreviated. See function metabias
backtransf	A logical indicating whether results for ratio of means (sm="ROM") should be back transformed in printouts and plots. If TRUE (default), results will be presented as ratio of means; otherwise log ratio of means will be shown.
title	Title of meta-analysis / systematic review.
complab	Comparison label.
outclab	Outcome label.
label.e	Label for experimental group.
label.c	Label for control group.
label.left	Graph label on left side of forest plot.
label.right	Graph label on right side of forest plot.
sm	A character string indicating which summary measure ("MD", "SMD", or "ROM") is to be used for pooling of studies.
pooledvar	A logical indicating if a pooled variance should be used for the mean difference (sm="MD").
method.smd	A character string indicating which method is used to estimate the standardised mean difference (sm="SMD"). Either "Hedges" for Hedges' g (default), "Cohen" for Cohen's d, or "Glass" for Glass' delta, can be abbreviated.

sd.glass	A character string indicating which standard deviation is used in the denominator for Glass' method to estimate the standardised mean difference. Either "control" using the standard deviation in the control group (sd.c) or "experimental" using the standard deviation in the experimental group (sd.e), can be abbreviated.
exact.smd	A logical indicating whether exact formulae should be used in estimation of the standardised mean difference and its standard error (see Details).
byvar	An optional vector containing grouping information (must be of same length as n.e).
bylab	A character string with a label for the grouping variable.
print.byvar	A logical indicating whether the name of the grouping variable should be printed in front of the group labels.
byseparator	A character string defining the separator between label and levels of grouping variable.
keepdata	A logical indicating whether original data (set) should be kept in meta object.
warn	A logical indicating whether warnings should be printed (e.g., if studies are excluded from meta-analysis due to zero standard deviations).
control	An optional list to control the iterative process to estimate the between-study variance tau^2. This argument is passed on to rma.uni.

Details

Calculation of fixed and random effects estimates for meta-analyses with continuous outcome data; inverse variance weighting is used for pooling.

Three different types of summary measures are available for continuous outcomes:

- mean difference (argument sm="MD")
- standardised mean difference (sm="SMD")
- ratio of means (sm="ROM")

Meta-analysis of ratio of means – also called response ratios – is described in Hedges et al. (1999) and Friedrich et al. (2008).

For the standardised mean difference three methods are implemented:

- Hedges' g (default, method.smd="Hedges") - see Hedges (1981)
- Cohen's d (method.smd="Cohen") - see Cohen (1988)
- Glass' delta (method.smd="Glass") - see Glass (1976)

Hedges (1981) calculated the exact bias in Cohen's d which is a ratio of gamma distributions with the degrees of freedom, i.e. total sample size minus two, as argument. By default (argument exact.smd=FALSE), an accurate approximation of this bias provided in Hedges (1981) is utilised for Hedges' g as well as its standard error; these approximations are also used in RevMan 5. Following Borenstein et al. (2009) these approximations are not used in the estimation of Cohen's d. White and Thomas (2005) argued that approximations are unnecessary with modern software and accordingly promote to use the exact formulae; this is possible using argument exact.smd=TRUE. For Hedges' g the exact formulae are used to calculate the standardised mean difference as well as

the standard error; for Cohen's d the exact formula is only used to calculate the standard error. In typical applications (with sample sizes above 10), the differences between using the exact formulae and the approximation will be minimal.

For Glass' delta, by default (argument `sd.glass="control"`), the standard deviation in the control group (`sd.c`) is used in the denominator of the standard mean difference. The standard deviation in the experimental group (`sd.e`) can be used by specifying `sd.glass="experimental"`.

Calculations are conducted on the log scale for ratio of means (`sm="ROM"`). Accordingly, list elements `TE`, `TE.fixed`, and `TE.random` contain the logarithm of ratio of means. In printouts and plots these values are back transformed if argument `backtransf=TRUE`.

For several arguments defaults settings are utilised (assignments using `gs` function). These defaults can be changed using the `settings.meta` function.

Internally, both fixed effect and random effects models are calculated regardless of values choosen for arguments `comb.fixed` and `comb.random`. Accordingly, the estimate for the random effects model can be extracted from component `TE.random` of an object of class `"meta"` even if argument `comb.random=FALSE`. However, all functions in R package **meta** will adequately consider the values for `comb.fixed` and `comb.random`. E.g. function `print.meta` will not print results for the random effects model if `comb.random=FALSE`.

The function `metagen` is called internally to calculate individual and overall treatment estimates and standard errors.

A prediction interval for treatment effect of a new study is calculated (Higgins et al., 2009) if arguments `prediction` and `comb.random` are `TRUE`.

R function `update.meta` can be used to redo the meta-analysis of an existing metacont object by only specifying arguments which should be changed.

For the random effects, the method by Hartung and Knapp (2003) is used to adjust test statistics and confidence intervals if argument `hakn=TRUE`.

The DerSimonian-Laird estimate (1986) is used in the random effects model if `method.tau="DL"`. The iterative Paule-Mandel method (1982) to estimate the between-study variance is used if argument `method.tau="PM"`. Internally, R function `paulemandel` is called which is based on R function `mpaule.default` from R package **metRology** from S.L.R. Ellison <s.ellison at lgc.co.uk>.

If R package **metafor** (Viechtbauer 2010) is installed, the following methods to estimate the between-study variance τ^2 (argument `method.tau`) are also available:

- Restricted maximum-likelihood estimator (`method.tau="REML"`)
- Maximum-likelihood estimator (`method.tau="ML"`)
- Hunter-Schmidt estimator (`method.tau="HS"`)
- Sidik-Jonkman estimator (`method.tau="SJ"`)
- Hedges estimator (`method.tau="HE"`)
- Empirical Bayes estimator (`method.tau="EB"`).

For these methods the R function `rma.uni` of R package **metafor** is called internally. See help page of R function `rma.uni` for more details on these methods to estimate between-study variance.

Value

An object of class `c("metacont", "meta")` with corresponding `print`, `summary`, and `forest` functions. The object is a list containing the following components:

`n.e, mean.e, sd.e,`

`n.c, mean.c, sd.c,`

`studlab, exclude, sm, level, level.comb,`

`comb.fixed, comb.random,`

`pooledvar, method.smd, sd.glass,`

`hakn, method.tau, tau.preset, TE.tau, method.bias,`

`tau.common, title, complab, outclab,`

`label.e, label.c, label.left, label.right,`

`byvar, bylab, print.byvar, byseparator, warn`
: As defined above.

`TE, seTE`
: Estimated treatment effect and standard error of individual studies.

`lower, upper`
: Lower and upper confidence interval limits for individual studies.

`zval, pval`
: z-value and p-value for test of treatment effect for individual studies.

`w.fixed, w.random`
: Weight of individual studies (in fixed and random effects model).

`TE.fixed, seTE.fixed`
: Estimated overall treatment effect and standard error (fixed effect model).

`lower.fixed, upper.fixed`
: Lower and upper confidence interval limits (fixed effect model).

`zval.fixed, pval.fixed`
: z-value and p-value for test of overall treatment effect (fixed effect model).

`TE.random, seTE.random`
: Estimated overall treatment effect and standard error (random effects model).

`lower.random, upper.random`
: Lower and upper confidence interval limits (random effects model).

`zval.random, pval.random`
: z-value or t-value and corresponding p-value for test of overall treatment effect (random effects model).

`prediction, level.predict`
: As defined above.

`seTE.predict`
: Standard error utilised for prediction interval.

`lower.predict, upper.predict`
: Lower and upper limits of prediction interval.

k — Number of studies combined in meta-analysis.

Q — Heterogeneity statistic.

tau — Square-root of between-study variance.

se.tau — Standard error of square-root of between-study variance.

C — Scaling factor utilised internally to calculate common tau-squared across subgroups.

method — Pooling method: "Inverse".

df.hakn — Degrees of freedom for test of treatment effect for Hartung-Knapp method (only if hakn=TRUE).

bylevs — Levels of grouping variable - if byvar is not missing.

TE.fixed.w, seTE.fixed.w — Estimated treatment effect and standard error in subgroups (fixed effect model) - if byvar is not missing.

lower.fixed.w, upper.fixed.w — Lower and upper confidence interval limits in subgroups (fixed effect model) - if byvar is not missing.

zval.fixed.w, pval.fixed.w — z-value and p-value for test of treatment effect in subgroups (fixed effect model) - if byvar is not missing.

TE.random.w, seTE.random.w — Estimated treatment effect and standard error in subgroups (random effects model) - if byvar is not missing.

lower.random.w, upper.random.w — Lower and upper confidence interval limits in subgroups (random effects model) - if byvar is not missing.

zval.random.w, pval.random.w — z-value or t-value and corresponding p-value for test of treatment effect in subgroups (random effects model) - if byvar is not missing.

w.fixed.w, w.random.w — Weight of subgroups (in fixed and random effects model) - if byvar is not missing.

df.hakn.w — Degrees of freedom for test of treatment effect for Hartung-Knapp method in subgroups - if byvar is not missing and hakn=TRUE.

n.harmonic.mean.w — Harmonic mean of number of observations in subgroups (for back transformation of Freeman-Tukey Double arcsine transformation) - if byvar is not missing.

n.e.w — Number of observations in experimental group in subgroups - if byvar is not missing.

n.c.w — Number of observations in control group in subgroups - if byvar is not missing.

k.w — Number of studies combined within subgroups - if byvar is not missing.

k.all.w — Number of all studies in subgroups - if byvar is not missing.

Q.w.fixed — Overall within subgroups heterogeneity statistic Q (based on fixed effect model) - if byvar is not missing.

`Q.w.random` Overall within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing (only calculated if argument `tau.common` is TRUE).

`df.Q.w` Degrees of freedom for test of overall within subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.w.fixed` P-value of within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.w.random` P-value of within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`Q.b.fixed` Overall between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.b.random` Overall between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`df.Q.b` Degrees of freedom for test of overall between subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.b.fixed` P-value of between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.b.random` P-value of between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`tau.w` Square-root of between-study variance within subgroups - if `byvar` is not missing.

`C.w` Scaling factor utilised internally to calculate common tau-squared across subgroups - if `byvar` is not missing.

`H.w` Heterogeneity statistic H within subgroups - if `byvar` is not missing.

`lower.H.w, upper.H.w` Lower and upper confidence limti for heterogeneity statistic H within subgroups - if `byvar` is not missing.

`I2.w` Heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`lower.I2.w, upper.I2.w` Lower and upper confidence limti for heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`keepdata` As defined above.

`data` Original data (set) used in function call (if `keepdata=TRUE`).

`subset` Information on subset of original data used in meta-analysis (if `keepdata=TRUE`).

`call` Function call.

`version` Version of R package **meta** used to create object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Borenstein et al. (2009), *Introduction to Meta-Analysis*, Chichester: Wiley.

Cohen J (1988), *Statistical Power Analysis for the Behavioral Sciences (second ed.)*, Lawrence Erlbaum Associates.

Cooper H & Hedges LV (1994), *The Handbook of Research Synthesis*. Newbury Park, CA: Russell Sage Foundation.

DerSimonian R & Laird N (1986), Meta-analysis in clinical trials. *Controlled Clinical Trials*, **7**, 177–88.

Friedrich JO, Adhikari NK, Beyene J (2008), The ratio of means method as an alternative to mean differences for analyzing continuous outcome variables in meta-analysis: A simulation study. *BMC Med Res Methodol*, **8**, 32.

Glass G (1976), Primary, secondary, and meta-analysis of research. *Educational Researcher*, **5**, 3–8.

Hartung J & Knapp G (2001), On tests of the overall treatment effect in meta-analysis with normally distributed responses. *Statistics in Medicine*, **20**, 1771–82. doi: 10.1002/sim.791 .

Hedges LV (1981), Distribution theory for Glass's estimator of effect size and related estimators. *Journal of Educational and Behavioral Statistics*, **6**, 107–28.

Hedges LV, Gurevitch J, Curtis PS (1999), The meta-analysis of response ratios in experimental ecology. *Ecology*, **80**, 1150–6.

Higgins JPT, Thompson SG, Spiegelhalter DJ (2009), A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society: Series A*, **172**, 137–59.

Knapp G & Hartung J (2003), Improved Tests for a Random Effects Meta-regression with a Single Covariate. *Statistics in Medicine*, **22**, 2693–710, doi: 10.1002/sim.1482 .

Paule RC & Mandel J (1982), Consensus values and weighting factors. *Journal of Research of the National Bureau of Standards*, **87**, 377–85.

Review Manager (RevMan) [Computer program]. Version 5.3. Copenhagen: The Nordic Cochrane Centre, The Cochrane Collaboration, 2014.

Viechtbauer W (2010), Conducting Meta-Analyses in R with the Metafor Package. *Journal of Statistical Software*, **36**, 1–48.

White IR, Thomas J (2005), Standardized mean differences in individually-randomized and cluster-randomized trials, with applications to meta-analysis. *Clinical Trials*, **2**, 141–51.

See Also

`update.meta`, `metabin`, `metagen`

Examples

```
data(Fleiss93cont)
meta1 <- metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c, data=Fleiss93cont, sm="SMD")
meta1
forest(meta1)

meta2 <- metacont(Fleiss93cont$n.e, Fleiss93cont$mean.e,
```

```
                  Fleiss93cont$sd.e,
                  Fleiss93cont$n.c, Fleiss93cont$mean.c,
                  Fleiss93cont$sd.c,
                  sm="SMD")
meta2

data(amlodipine)
meta3 <- metacont(n.amlo, mean.amlo, sqrt(var.amlo),
                  n.plac, mean.plac, sqrt(var.plac),
                  data=amlodipine, studlab=study)
summary(meta3)

# Use pooled variance
#
meta4 <- metacont(n.amlo, mean.amlo, sqrt(var.amlo),
                  n.plac, mean.plac, sqrt(var.plac),
                  data=amlodipine, studlab=study,
                  pooledvar=TRUE)
summary(meta4)

# Use Cohen's d instead of Hedges' g as effect measure
#
meta5 <- metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c, data=Fleiss93cont,
                  sm="SMD", method.smd="Cohen")
meta5

# Use Glass' delta instead of Hedges' g as effect measure
#
meta6 <- metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c, data=Fleiss93cont,
                  sm="SMD", method.smd="Glass")
meta6

# Use Glass' delta based on the standard deviation in the experimental group
#
meta7 <- metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c, data=Fleiss93cont,
                  sm="SMD", method.smd="Glass", sd.glass="experimental")
meta7

# Calculate Hedges' g based on exact formulae
#
update(meta1, exact.smd=TRUE)

#
# Meta-analysis of response ratios (Hedges et al., 1999)
#
data(woodyplants)
meta8 <- metacont(n.elev, mean.elev, sd.elev,
  n.amb, mean.amb, sd.amb,
                  data=woodyplants, sm="ROM")
summary(meta8)
summary(meta8, backtransf=FALSE)
```

metacor *Meta-analysis of correlations*

Description

Calculation of fixed and random effects estimates for meta-analyses with correlations; inverse variance weighting is used for pooling.

Usage

```
metacor(cor, n, studlab,
        data=NULL, subset=NULL, exclude=NULL,
        sm=gs("smcor"),
        level=gs("level"), level.comb=gs("level.comb"),
        comb.fixed=gs("comb.fixed"), comb.random=gs("comb.random"),
        hakn=gs("hakn"),
        method.tau=gs("method.tau"), tau.preset=NULL, TE.tau=NULL,
        tau.common=gs("tau.common"),
        prediction=gs("prediction"), level.predict=gs("level.predict"),
null.effect=0,
        method.bias=gs("method.bias"),
        backtransf=gs("backtransf"),
        title=gs("title"), complab=gs("complab"), outclab="",
        byvar, bylab, print.byvar=gs("print.byvar"),
        byseparator = gs("byseparator"),
        keepdata=gs("keepdata"),
        control=NULL
        )
```

Arguments

cor	Correlation.
n	Number of observations.
studlab	An optional vector with study labels.
data	An optional data frame containing the study information, i.e., cor and n.
subset	An optional vector specifying a subset of studies to be used.
exclude	An optional vector specifying studies to exclude from meta-analysis, however, to include in printouts and forest plots.
sm	A character string indicating which summary measure ("ZCOR" or "COR") is to be used for pooling of studies.
level	The level used to calculate confidence intervals for individual studies.
level.comb	The level used to calculate confidence intervals for pooled estimates.
comb.fixed	A logical indicating whether a fixed effect meta-analysis should be conducted.

comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
prediction	A logical indicating whether a prediction interval should be printed.
level.predict	The level used to calculate prediction interval for a new study.
hakn	A logical indicating whether the method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
method.tau	A character string indicating which method is used to estimate the between-study variance τ^2. Either "DL", "PM", "REML", "ML", "HS", "SJ", "HE", or "EB", can be abbreviated.
tau.preset	Prespecified value for the square-root of the between-study variance τ^2.
TE.tau	Overall treatment effect used to estimate the between-study variance tau-squared.
tau.common	A logical indicating whether tau-squared should be the same across subgroups.
null.effect	A numeric value specifying the effect under the null hypothesis.
method.bias	A character string indicating which test is to be used. Either "rank", "linreg", or "mm", can be abbreviated. See function metabias
backtransf	A logical indicating whether results for Fisher's z transformed correlations (sm="ZCOR") should be back transformed in printouts and plots. If TRUE (default), results will be presented as correlations; otherwise Fisher's z transformed correlations will be shown.
title	Title of meta-analysis / systematic review.
complab	Comparison label.
outclab	Outcome label.
byvar	An optional vector containing grouping information (must be of same length as event.e).
bylab	A character string with a label for the grouping variable.
print.byvar	A logical indicating whether the name of the grouping variable should be printed in front of the group labels.
byseparator	A character string defining the separator between label and levels of grouping variable.
keepdata	A logical indicating whether original data (set) should be kept in meta object.
control	An optional list to control the iterative process to estimate the between-study variance tau^2. This argument is passed on to rma.uni.

Details

Fixed effect and random effects meta-analysis of correlations based either on Fisher's z transformation of correlations (sm="ZCOR") or direct combination of correlations (sm="COR") (see Cooper et al., p264-5 and p273-4).

Only few statisticians would advocate the use of untransformed correlations unless sample sizes are very large (see Cooper et al., p265). The artificial example given below shows that the smallest study gets the largest weight if correlations are combined directly because the correlation is closest to 1.

For several arguments defaults settings are utilised (assignments using gs function). These defaults can be changed using the settings.meta function.

Internally, both fixed effect and random effects models are calculated regardless of values choosen for arguments comb.fixed and comb.random. Accordingly, the estimate for the random effects model can be extracted from component TE.random of an object of class "meta" even if argument comb.random=FALSE. However, all functions in R package **meta** will adequately consider the values for comb.fixed and comb.random. E.g. function print.meta will not print results for the random effects model if comb.random=FALSE.

A prediction interval for treatment effect of a new study is calculated (Higgins et al., 2009) if arguments prediction and comb.random are TRUE.

R function update.meta can be used to redo the meta-analysis of an existing metacor object by only specifying arguments which should be changed.

For the random effects, the method by Hartung and Knapp (2003) is used to adjust test statistics and confidence intervals if argument hakn=TRUE.

The DerSimonian-Laird estimate (1986) is used in the random effects model if method.tau="DL". The iterative Paule-Mandel method (1982) to estimate the between-study variance is used if argument method.tau="PM". Internally, R function paulemandel is called which is based on R function mpaule.default from R package **metRology** from S.L.R. Ellison <s.ellison at lgc.co.uk>.

If R package **metafor** (Viechtbauer 2010) is installed, the following methods to estimate the between-study variance τ^2 (argument method.tau) are also available:

- Restricted maximum-likelihood estimator (method.tau="REML")
- Maximum-likelihood estimator (method.tau="ML")
- Hunter-Schmidt estimator (method.tau="HS")
- Sidik-Jonkman estimator (method.tau="SJ")
- Hedges estimator (method.tau="HE")
- Empirical Bayes estimator (method.tau="EB").

For these methods the R function rma.uni of R package **metafor** is called internally. See help page of R function rma.uni for more details on these methods to estimate between-study variance.

Value

An object of class c("metacor", "meta") with corresponding print, summary, and forest functions. The object is a list containing the following components:

cor, n, studlab, exclude,

sm, level, level.comb,

comb.fixed, comb.random,

hakn, method.tau, tau.preset, TE.tau, null.effect,

method.bias, tau.common, title, complab, outclab,

`byvar, bylab, print.byvar, byseparator`
: As defined above.

`TE, seTE`
: Either Fisher's z transformation of correlations (`sm="ZCOR"`) or correlations (`sm="COR"`) for individual studies.

`lower, upper`
: Lower and upper confidence interval limits for individual studies.

`zval, pval`
: z-value and p-value for test of treatment effect for individual studies.

`w.fixed, w.random`
: Weight of individual studies (in fixed and random effects model).

`TE.fixed, seTE.fixed`
: Estimated overall effect (Fisher's z transformation of correlation or correlation) and its standard error (fixed effect model).

`lower.fixed, upper.fixed`
: Lower and upper confidence interval limits (fixed effect model).

`zval.fixed, pval.fixed`
: z-value and p-value for test of overall effect (fixed effect model).

`TE.random, seTE.random`
: Estimated overall effect (Fisher's z transformation of correlation or correlation) and its standard error (random effects model).

`lower.random, upper.random`
: Lower and upper confidence interval limits (random effects model).

`zval.random, pval.random`
: z-value or t-value and corresponding p-value for test of overall effect (random effects model).

`prediction, level.predict`
: As defined above.

`seTE.predict`
: Standard error utilised for prediction interval.

`lower.predict, upper.predict`
: Lower and upper limits of prediction interval.

`k`
: Number of studies combined in meta-analysis.

`Q`
: Heterogeneity statistic Q.

`tau`
: Square-root of between-study variance.

`se.tau`
: Standard error of square-root of between-study variance.

`C`
: Scaling factor utilised internally to calculate common tau-squared across subgroups.

`method`
: A character string indicating method used for pooling: `"Inverse"`

`df.hakn`
: Degrees of freedom for test of treatment effect for Hartung-Knapp method (only if `hakn=TRUE`).

`bylevs`
: Levels of grouping variable - if `byvar` is not missing.

`TE.fixed.w, seTE.fixed.w`
: Estimated treatment effect and standard error in subgroups (fixed effect model) - if `byvar` is not missing.

`lower.fixed.w, upper.fixed.w`
: Lower and upper confidence interval limits in subgroups (fixed effect model) - if `byvar` is not missing.

`zval.fixed.w, pval.fixed.w`
: z-value and p-value for test of treatment effect in subgroups (fixed effect model) - if `byvar` is not missing.

`TE.random.w, seTE.random.w`
: Estimated treatment effect and standard error in subgroups (random effects model) - if `byvar` is not missing.

`lower.random.w, upper.random.w`
: Lower and upper confidence interval limits in subgroups (random effects model) - if `byvar` is not missing.

`zval.random.w, pval.random.w`
: z-value or t-value and corresponding p-value for test of treatment effect in subgroups (random effects model) - if `byvar` is not missing.

`w.fixed.w, w.random.w`
: Weight of subgroups (in fixed and random effects model) - if `byvar` is not missing.

`df.hakn.w`
: Degrees of freedom for test of treatment effect for Hartung-Knapp method in subgroups - if `byvar` is not missing and `hakn=TRUE`.

`n.harmonic.mean.w`
: Harmonic mean of number of observations in subgroups (for back transformation of Freeman-Tukey Double arcsine transformation) - if `byvar` is not missing.

`n.w`
: Number of observations in subgroups - if `byvar` is not missing.

`k.w`
: Number of studies combined within subgroups - if `byvar` is not missing.

`k.all.w`
: Number of all studies in subgroups - if `byvar` is not missing.

`Q.w.fixed`
: Overall within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.w.random`
: Overall within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing (only calculated if argument `tau.common` is TRUE).

`df.Q.w`
: Degrees of freedom for test of overall within subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.w.fixed`
: P-value of within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.w.random`
: P-value of within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`Q.b.fixed`
: Overall between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.b.random`
: Overall between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`df.Q.b`
: Degrees of freedom for test of overall between subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.b.fixed`
: P-value of between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.b.random`
: P-value of between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`tau.w`
: Square-root of between-study variance within subgroups - if `byvar` is not missing.

`C.w`
: Scaling factor utilised internally to calculate common tau-squared across subgroups - if `byvar` is not missing.

`H.w`
: Heterogeneity statistic H within subgroups - if `byvar` is not missing.

`lower.H.w, upper.H.w`
: Lower and upper confidence limti for heterogeneity statistic H within subgroups - if `byvar` is not missing.

`I2.w`
: Heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`lower.I2.w, upper.I2.w`
: Lower and upper confidence limti for heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`keepdata`
: As defined above.

`data`
: Original data (set) used in function call (if `keepdata=TRUE`).

`subset`
: Information on subset of original data used in meta-analysis (if `keepdata=TRUE`).

`call`
: Function call.

`version`
: Version of R package **meta** used to create object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Cooper H, Hedges LV, Valentine JC (2009), *The Handbook of Research Synthesis and Meta-Analysis*, 2nd Edition. New York: Russell Sage Foundation.

DerSimonian R & Laird N (1986), Meta-analysis in clinical trials. *Controlled Clinical Trials*, **7**, 177–188.

Higgins JPT, Thompson SG, Spiegelhalter DJ (2009), A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society: Series A*, **172**, 137–159.

Knapp G & Hartung J (2003), Improved Tests for a Random Effects Meta-regression with a Single Covariate. *Statistics in Medicine*, **22**, 2693–710, doi: 10.1002/sim.1482 .

Paule RC & Mandel J (1982), Consensus values and weighting factors. *Journal of Research of the National Bureau of Standards*, **87**, 377–385.

Viechtbauer W (2010), Conducting Meta-Analyses in R with the Metafor Package. *Journal of Statistical Software*, **36**, 1–48.

See Also

`update.meta`, `metacont`, `metagen`, `print.meta`

Examples

```
m1 <- metacor(c(0.85, 0.7, 0.95), c(20, 40, 10))

#
# Print correlations (back transformed from Fisher's z transformation)
#
m1

#
# Print Fisher's z transformed correlations
#
print(m1, backtransf=FALSE)

#
# Forest plot with back transformed correlations
#
forest(m1)

#
# Forest plot with Fisher's z transformed correlations
#
forest(m1, backtransf=FALSE)


m2 <- update(m1, sm="cor")
m2
# Identical forest plots (as back transformation is the identity transformation)
# forest(m2)
# forest(m2, backtransf=FALSE)
```

`metacr` *Meta-analysis of outcome data from Cochrane review*

Description

Wrapper function to perform meta-analysis for a single outcome of a Cochrane Intervention review.

Usage

```
metacr(x, comp.no=1, outcome.no=1,
       method, sm,
       level=gs("level"), level.comb=gs("level.comb"),
       comb.fixed, comb.random,
       hakn=FALSE,
       method.tau="DL",
       tau.common=FALSE,
       prediction=gs("prediction"), level.predict=gs("level.predict"),
       swap.events, logscale,
       backtransf=gs("backtransf"),
```

```
        title, complab, outclab,
        keepdata=gs("keepdata"), warn=FALSE)
```

Arguments

x	An object of class rm5 created by R function read.rm5.
comp.no	Comparison number.
outcome.no	Outcome number.
method	A character string indicating which method is to be used for pooling of studies. One of "Inverse", "MH", or "Peto", can be abbreviated.
sm	A character string indicating which summary measure ("RR", "OR", "RD", "ASD", "HR", "MD", or "SMD", or "ROM") is to be used for pooling of studies.
level	The level used to calculate confidence intervals for individual studies.
level.comb	The level used to calculate confidence intervals for pooled estimates.
comb.fixed	A logical indicating whether a fixed effect meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
hakn	A logical indicating whether the method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
method.tau	A character string indicating which method is used to estimate the between-study variance τ^2. Either "DL", "PM", "REML", "ML", "HS", "SJ", "HE", or "EB", can be abbreviated.
tau.common	A logical indicating whether tau-squared should be the same across subgroups.
prediction	A logical indicating whether a prediction interval should be printed.
level.predict	The level used to calculate prediction interval for a new study.
swap.events	A logical indicating whether events and non-events should be interchanged.
logscale	A logical indicating whether effect estimates are entered on log-scale.
backtransf	A logical indicating whether results should be back transformed in printouts and plots. If backtransf=TRUE (default), results for sm="OR" are printed as odds ratios rather than log odds ratios and results for sm="ZCOR" are printed as correlations rather than Fisher's z transformed correlations, for example.
title	Title of meta-analysis / systematic review.
complab	Comparison label.
outclab	Outcome label.
keepdata	A logical indicating whether original data (set) should be kept in meta object.
warn	A logical indicating whether warnings should be printed (e.g., if incr is added to studies with zero cell frequencies).

Details

Cochrane Intervention reviews are based on the comparison of two interventions. Each Cochrane Intervention review can have a variable number of comparisons. For each comparison, a variable number of outcomes can be define. For each outcome, a seperate meta-analysis is conducted. Review Manager 5 (RevMan 5) is the current software used for preparing and maintaining Cochrane Reviews (http://community.cochrane.org/tools/review-production-tools/revman-5).

This wrapper function can be used to perform meta-analysis for a single outcome of a Cochrane Intervention review. Internally, R functions `metabin`, `metacont`, and `metagen` are called - depending on the definition of the outcome in RevMan 5.

Note, it is recommended to specify the RevMan 5 before executing `metacr`, i.e.,

```
settings.meta("revman5")
```

Value

An object of class `"meta"` and `"metabin"`, `"metacont"`, or `"metagen"` depending on outcome type utilised in Cochrane Intervention review for selected outcome.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Review Manager (RevMan) [Computer program]. Version 5.3. Copenhagen: The Nordic Cochrane Centre, The Cochrane Collaboration, 2014.

See Also

metabin, metacont, metagen, read.rm5, settings.meta

Examples

```
# Locate export data file "Fleiss93_CR.csv"
# in sub-directory of package "meta"
#
filename <- system.file("data/Fleiss93_CR.csv.gz", package = "meta")
#
Fleiss93_CR <- read.rm5(filename)

# Choose RevMan 5 settings and store old settings
#
oldset <- settings.meta("revman5")

# Same result as R command example(Fleiss93):
#
metacr(Fleiss93_CR)

# Same result as R command example(Fleiss93cont):
#
```

```
metacr(Fleiss93_CR, 1, 2)

forest(metacr(Fleiss93_CR, 1, 2))

# Change summary measure to RR
#
m1 <- metacr(Fleiss93_CR)
update(m1, sm="RR")

# Use old settings
#
settings.meta(oldset)
```

`metacum` *Cumulative meta-analysis*

Description

Performs a cumulative meta-analysis.

Usage

```
metacum(x, pooled, sortvar)
```

Arguments

`x`	An object of class `meta`.
`pooled`	A character string indicating whether a fixed effect or random effects model is used for pooling. Either missing (see Details), `"fixed"`, or `"random"`, can be abbreviated.
`sortvar`	An optional vector used to sort the individual studies (must be of same length as `x$TE`).

Details

A cumulative meta-analysis is performed. Studies are included sequentially as defined by `sortvar`.

Information from object `x` is utilised if argument `pooled` is missing. A fixed effect model is assumed (`pooled="fixed"`) if argument `x$comb.fixed` is `TRUE`; a random effects model is assumed (`pooled="random"`) if argument `x$comb.random` is `TRUE` and `x$comb.fixed` is `FALSE`.

Value

An object of class `c("metacum", "meta")` with corresponding `print`, and `forest` functions. The object is a list containing the following components:

`TE, seTE`	Estimated treatment effect and standard error of pooled estimate in cumulative meta-analyses.
`lower, upper`	Lower and upper confidence interval limits.

studlab	Study label describing addition of studies.
p.value	P-value for test of overall effect.
w	Sum of weights from fixed effect or random effects model.
I2	Heterogeneity statistic I2.
Rb	Heterogeneity statistic Rb.
tau	Square-root of between-study variance.
df.hakn	Degrees of freedom for test of treatment effect for Hartung-Knapp method (only if hakn=TRUE).
sm	Summary measure.
method	Method used for pooling.
k	Number of studies combined in meta-analysis.
pooled	As defined above.
comb.fixed	A logical indicating whether analysis is based on fixed effect model.
comb.random	A logical indicating whether analysis is based on random effects model.
TE.fixed, seTE.fixed	Value is NA.
TE.random, seTE.random	Value is NA.
Q	Value is NA.
level.comb	The level used to calculate confidence intervals for pooled estimates.
hakn	A logical indicating whether the method by Hartung and Knapp is used to adjust test statistics and confidence intervals.
method.tau	A character string indicating which method is used to estimate the between-study variance τ^2.
tau.preset	Prespecified value for the square-root of the between-study variance τ^2.
TE.tau	Overall treatment effect used to estimate the between-study variance τ^2.
n.harmonic.mean	Harmonic mean of number of observations (for back transformation of Freeman-Tukey Double arcsine transformation).
version	Version of R package **meta** used to create object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Cooper H & Hedges LV (1994), *The Handbook of Research Synthesis*. Newbury Park, CA: Russell Sage Foundation.

See Also

metabin, metacont, print.meta

Examples

```
data(Fleiss93)
meta1 <- metabin(event.e, n.e, event.c, n.c,
                 data=Fleiss93, studlab=study,
                 sm="RR", method="I")
meta1

metacum(meta1)
metacum(meta1, pooled="random")

forest(metacum(meta1))
forest(metacum(meta1, pooled="random"))

metacum(meta1, sortvar=study)
metacum(meta1, sortvar=7:1)

meta2 <- update(meta1, title="Fleiss93 meta-analysis",
                backtransf=FALSE)
metacum(meta2)

data(Fleiss93cont)
meta3 <- metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c,
                  data = Fleiss93cont, sm = "SMD")
metacum(meta3)
```

`metagen` *Generic inverse variance meta-analysis*

Description

Fixed and random effects meta-analysis based on estimates (e.g. log hazard ratios) and their standard errors; inverse variance weighting is used for pooling.

Usage

```
metagen(TE, seTE, studlab,
        data=NULL, subset=NULL, exclude=NULL, sm="",
        level=gs("level"), level.comb=gs("level.comb"),
        comb.fixed=gs("comb.fixed"), comb.random=gs("comb.random"),
        hakn=gs("hakn"),
        method.tau=gs("method.tau"), tau.preset=NULL, TE.tau=NULL,
        tau.common=gs("tau.common"),
        prediction=gs("prediction"), level.predict=gs("level.predict"),
        null.effect=0,
        method.bias=gs("method.bias"),
        n.e=NULL, n.c=NULL,
        backtransf=gs("backtransf"),
        pscale=1, irscale = 1, irunit = "person-years",
```

```
title=gs("title"), complab=gs("complab"), outclab="",
label.e=gs("label.e"), label.c=gs("label.c"),
label.left=gs("label.left"), label.right=gs("label.right"),
byvar, bylab, print.byvar=gs("print.byvar"),
byseparator = gs("byseparator"),
keepdata=gs("keepdata"),
warn=gs("warn"), control=NULL)
```

Arguments

TE	Estimate of treatment effect, e.g., log hazard ratio or risk difference.
seTE	Standard error of treatment estimate.
studlab	An optional vector with study labels.
data	An optional data frame containing the study information.
subset	An optional vector specifying a subset of studies to be used.
exclude	An optional vector specifying studies to exclude from meta-analysis, however, to include in printouts and forest plots.
sm	A character string indicating underlying summary measure, e.g., "RD", "RR", "OR", "ASD", "HR", "MD", "SMD", or "ROM".
level	The level used to calculate confidence intervals for individual studies.
level.comb	The level used to calculate confidence intervals for pooled estimates.
comb.fixed	A logical indicating whether a fixed effect meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
prediction	A logical indicating whether a prediction interval should be printed.
level.predict	The level used to calculate prediction interval for a new study.
null.effect	A numeric value specifying the effect under the null hypothesis.
n.e	Number of observations in experimental group.
n.c	Number of observations in control group.
hakn	A logical indicating whether method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
method.tau	A character string indicating which method is used to estimate the between-study variance τ^2. Either "DL", "PM", "REML", "ML", "HS", "SJ", "HE", or "EB", can be abbreviated.
tau.preset	Prespecified value for the square-root of the between-study variance τ^2.
TE.tau	Overall treatment effect used to estimate the between-study variance tau-squared.
tau.common	A logical indicating whether tau-squared should be the same across subgroups.
method.bias	A character string indicating which test is to be used. Either "rank", "linreg", or "mm", can be abbreviated. See function metabias
backtransf	A logical indicating whether results should be back transformed in printouts and plots. If backtransf=TRUE (default), results for sm="OR" are printed as odds ratios rather than log odds ratios and results for sm="ZCOR" are printed as correlations rather than Fisher's z transformed correlations, for example.

pscale	A numeric giving scaling factor for printing of single event probabilities or risk differences, i.e. if argument sm is equal to "PLOGIT", "PLN", "PRAW", "PAS", "PFT", or "RD".
irscale	A numeric defining a scaling factor for printing of single incidence rates or incidence rate differences, i.e. if argument sm is equal to "IR", "IRLN", "IRS", "IRFT", or "IRD".
irunit	A character specifying the time unit used to calculate rates, e.g. person-years.
title	Title of meta-analysis / systematic review.
complab	Comparison label.
outclab	Outcome label.
label.e	Label for experimental group.
label.c	Label for control group.
label.left	Graph label on left side of forest plot.
label.right	Graph label on right side of forest plot.
byvar	An optional vector containing grouping information (must be of same length as TE).
bylab	A character string with a label for the grouping variable.
print.byvar	A logical indicating whether the name of the grouping variable should be printed in front of the group labels.
byseparator	A character string defining the separator between label and levels of grouping variable.
keepdata	A logical indicating whether original data (set) should be kept in meta object.
warn	A logical indicating whether warnings should be printed (e.g., if studies are excluded from meta-analysis due to zero standard errors).
control	An optional list to control the iterative process to estimate the between-study variance tau^2. This argument is passed on to rma.uni.

Details

Generic method for meta-analysis, only treatment estimates and their standard error are needed. The method is useful, e.g., for pooling of survival data (using log hazard ratio and standard errors as input). The inverse variance method is used for pooling.

For several arguments defaults settings are utilised (assignments using gs function). These defaults can be changed using the settings.meta function.

Internally, both fixed effect and random effects models are calculated regardless of values choosen for arguments comb.fixed and comb.random. Accordingly, the estimate for the random effects model can be extracted from component TE.random of an object of class "meta" even if argument comb.random=FALSE. However, all functions in R package **meta** will adequately consider the values for comb.fixed and comb.random. E.g. function print.meta will not print results for the random effects model if comb.random=FALSE.

A prediction interval for treatment effect of a new study is calculated (Higgins et al., 2009) if arguments prediction and comb.random are TRUE.

R function update.meta can be used to redo the meta-analysis of an existing metagen object by only specifying arguments which should be changed.

For the random effects, the method by Hartung and Knapp (2003) is used to adjust test statistics and confidence intervals if argument hakn=TRUE.

The DerSimonian-Laird estimate (1986) is used in the random effects model if method.tau="DL". The iterative Paule-Mandel method (1982) to estimate the between-study variance is used if argument method.tau="PM". Internally, R function paulemandel is called which is based on R function mpaule.default from R package **metRology** from S.L.R. Ellison <s.ellison at lgc.co.uk>.

If R package **metafor** (Viechtbauer 2010) is installed, the following methods to estimate the between-study variance τ^2 (argument method.tau) are also available:

- Restricted maximum-likelihood estimator (method.tau="REML")
- Maximum-likelihood estimator (method.tau="ML")
- Hunter-Schmidt estimator (method.tau="HS")
- Sidik-Jonkman estimator (method.tau="SJ")
- Hedges estimator (method.tau="HE")
- Empirical Bayes estimator (method.tau="EB").

For these methods the R function rma.uni of R package **metafor** is called internally. See help page of R function rma.uni for more details on these methods to estimate between-study variance.

Argument pscale can be used to rescale single proportions or risk differences, e.g. pscale=1000 means that proportions are expressed as events per 1000 observations. This is useful in situations with (very) low event probabilities.

Argument irscale can be used to rescale single rates or rate differences, e.g. irscale=1000 means that rates are expressed as events per 1000 time units, e.g. person-years. This is useful in situations with (very) low rates. Argument irunit can be used to specify the time unit used in individual studies (default: "person-years"). This information is printed in summaries and forest plots if argument irscale is not equal to 1.

Value

An object of class c("metagen", "meta") with corresponding print, summary, and forest functions. The object is a list containing the following components:

TE, seTE, studlab, exclude, n.e, n.c

sm, level, level.comb,

comb.fixed, comb.random,

hakn, method.tau, tau.preset, TE.tau, method.bias,

tau.common, title, complab, outclab,

label.e, label.c, label.left, label.right,

byvar, bylab, print.byvar, byseparator, warn
: As defined above.

lower, upper
: Lower and upper confidence interval limits for individual studies.

zval, pval
: z-value and p-value for test of treatment effect for individual studies.

w.fixed, w.random
: Weight of individual studies (in fixed and random effects model).

TE.fixed, seTE.fixed
: Estimated overall treatment effect and standard error (fixed effect model).

lower.fixed, upper.fixed
: Lower and upper confidence interval limits (fixed effect model).

zval.fixed, pval.fixed
: z-value and p-value for test of overall treatment effect (fixed effect model).

TE.random, seTE.random
: Estimated overall treatment effect and standard error (random effects model).

lower.random, upper.random
: Lower and upper confidence interval limits (random effects model).

zval.random, pval.random
: z-value or t-value and corresponding p-value for test of overall treatment effect (random effects model).

prediction, level.predict
: As defined above.

seTE.predict
: Standard error utilised for prediction interval.

lower.predict, upper.predict
: Lower and upper limits of prediction interval.

null.effect
: As defined above.

k
: Number of studies combined in meta-analysis.

Q
: Heterogeneity statistic.

df.Q
: Degrees of freedom for heterogeneity statistic.

pval.Q
: P-value of heterogeneity test.

tau
: Square-root of between-study variance.

se.tau
: Standard error of square-root of between-study variance.

C
: Scaling factor utilised internally to calculate common tau-squared across subgroups.

method
: Pooling method: "Inverse".

df.hakn
: Degrees of freedom for test of treatment effect for Hartung-Knapp method (only if hakn=TRUE).

bylevs
: Levels of grouping variable - if byvar is not missing.

TE.fixed.w, seTE.fixed.w
: Estimated treatment effect and standard error in subgroups (fixed effect model) - if byvar is not missing.

lower.fixed.w, upper.fixed.w
: Lower and upper confidence interval limits in subgroups (fixed effect model) - if byvar is not missing.

`zval.fixed.w, pval.fixed.w`
: z-value and p-value for test of treatment effect in subgroups (fixed effect model) - if `byvar` is not missing.

`TE.random.w, seTE.random.w`
: Estimated treatment effect and standard error in subgroups (random effects model) - if `byvar` is not missing.

`lower.random.w, upper.random.w`
: Lower and upper confidence interval limits in subgroups (random effects model) - if `byvar` is not missing.

`zval.random.w, pval.random.w`
: z-value or t-value and corresponding p-value for test of treatment effect in subgroups (random effects model) - if `byvar` is not missing.

`w.fixed.w, w.random.w`
: Weight of subgroups (in fixed and random effects model) - if `byvar` is not missing.

`df.hakn.w`
: Degrees of freedom for test of treatment effect for Hartung-Knapp method in subgroups - if `byvar` is not missing and `hakn=TRUE`.

`n.harmonic.mean.w`
: Harmonic mean of number of observations in subgroups (for back transformation of Freeman-Tukey Double arcsine transformation) - if `byvar` is not missing.

`n.e.w`
: Number of observations in experimental group in subgroups - if `byvar` is not missing.

`n.c.w`
: Number of observations in control group in subgroups - if `byvar` is not missing.

`k.w`
: Number of studies combined within subgroups - if `byvar` is not missing.

`k.all.w`
: Number of all studies in subgroups - if `byvar` is not missing.

`Q.w.fixed`
: Overall within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.w.random`
: Overall within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing (only calculated if argument `tau.common` is TRUE).

`df.Q.w`
: Degrees of freedom for test of overall within subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.w.fixed`
: P-value of within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.w.random`
: P-value of within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`Q.b.fixed`
: Overall between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.b.random`
: Overall between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`df.Q.b`
: Degrees of freedom for test of overall between subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.b.fixed` P-value of between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.b.random` P-value of between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`tau.w` Square-root of between-study variance within subgroups - if `byvar` is not missing.

`C.w` Scaling factor utilised internally to calculate common tau-squared across subgroups - if `byvar` is not missing.

`H.w` Heterogeneity statistic H within subgroups - if `byvar` is not missing.

`lower.H.w, upper.H.w` Lower and upper confidence limti for heterogeneity statistic H within subgroups - if `byvar` is not missing.

`I2.w` Heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`lower.I2.w, upper.I2.w` Lower and upper confidence limti for heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`keepdata` As defined above.

`data` Original data (set) used in function call (if `keepdata=TRUE`).

`subset` Information on subset of original data used in meta-analysis (if `keepdata=TRUE`).

`call` Function call.

`version` Version of R package **meta** used to create object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Cooper H & Hedges LV (1994), *The Handbook of Research Synthesis*. Newbury Park, CA: Russell Sage Foundation.

DerSimonian R & Laird N (1986), Meta-analysis in clinical trials. *Controlled Clinical Trials*, **7**, 177–188.

Higgins JPT, Thompson SG, Spiegelhalter DJ (2009), A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society: Series A*, **172**, 137–159.

Knapp G & Hartung J (2003), Improved Tests for a Random Effects Meta-regression with a Single Covariate. *Statistics in Medicine*, **22**, 2693–2710, doi: 10.1002/sim.1482 .

Paule RC & Mandel J (1982), Consensus values and weighting factors. *Journal of Research of the National Bureau of Standards*, **87**, 377–385.

Viechtbauer W (2010), Conducting Meta-Analyses in R with the Metafor Package. *Journal of Statistical Software*, **36**, 1–48.

See Also

`update.meta`, `metabin`, `metacont`, `print.meta`

Examples

```
data(Fleiss93)
meta1 <- metabin(event.e, n.e, event.c, n.c, data=Fleiss93, sm="RR", method="I")
meta1

#
# Identical results by using the following commands:
#
meta1
metagen(meta1$TE, meta1$seTE, sm="RR")

forest(metagen(meta1$TE, meta1$seTE, sm="RR"))


#
# Meta-analysis with prespecified between-study variance
#
summary(metagen(meta1$TE, meta1$seTE, sm="RR", tau.preset=sqrt(0.1)))


#
# Meta-analysis of survival data:
#
logHR <- log(c(0.95, 1.5))
selogHR <- c(0.25, 0.35)

metagen(logHR, selogHR, sm="HR")


#
# Paule-Mandel method to estimate between-study variance
# Data from Paule & Mandel (1982)
#
average <- c(27.044, 26.022, 26.340, 26.787, 26.796)
variance <- c(0.003, 0.076, 0.464, 0.003, 0.014)
#
summary(metagen(average, sqrt(variance), sm="MD", method.tau="PM"))
```

`metainc` *Meta-analysis of incidence rates*

Description

Calculation of fixed effect and random effects estimates (incidence rate ratio or incidence rate difference) for meta-analyses with event counts. Mantel-Haenszel, Cochran, inverse variance method, and generalised linear mixed model (GLMM) are available for pooling. For GLMMs, the `rma.glmm` function from R package **metafor** (Viechtbauer 2010) is called internally.

Usage

```
metainc(event.e, time.e, event.c, time.c, studlab,
        data=NULL, subset=NULL, exclude=NULL, method="MH",
        sm=gs("sminc"),
        incr=gs("incr"), allincr=gs("allincr"),
        addincr=gs("addincr"),
        model.glmm = "UM.FS",
        level=gs("level"), level.comb=gs("level.comb"),
        comb.fixed=gs("comb.fixed"), comb.random=gs("comb.random"),
        hakn=gs("hakn"),
        method.tau=
        ifelse(!is.na(charmatch(tolower(method), "glmm", nomatch = NA)),
               "ML", gs("method.tau")),
        tau.preset=NULL, TE.tau=NULL,
        tau.common=gs("tau.common"),
        prediction=gs("prediction"), level.predict=gs("level.predict"),
        method.bias=gs("method.bias"),
        n.e=NULL, n.c=NULL,
        backtransf=gs("backtransf"), irscale = 1, irunit="person-years",
        title=gs("title"), complab=gs("complab"), outclab="",
        label.e=gs("label.e"), label.c=gs("label.c"),
        label.left=gs("label.left"), label.right=gs("label.right"),
        byvar, bylab, print.byvar=gs("print.byvar"),
        byseparator = gs("byseparator"),
        keepdata=gs("keepdata"),
        warn=gs("warn"),
        control=NULL,
...)
```

Arguments

event.e	Number of events in experimental group.
time.e	Person time at risk in experimental group.
event.c	Number of events in control group.
time.c	Person time at risk in control group.
studlab	An optional vector with study labels.
data	An optional data frame containing the study information, i.e., event.e, time.e, event.c, and time.c.
subset	An optional vector specifying a subset of studies to be used.
exclude	An optional vector specifying studies to exclude from meta-analysis, however, to include in printouts and forest plots.
method	A character string indicating which method is to be used for pooling of studies. One of "MH", "Inverse", "Cochran", or "GLMM" can be abbreviated.
sm	A character string indicating which summary measure ("IRR" or "IRD") is to be used for pooling of studies, see Details.

incr	A numerical value which is added to each cell frequency for studies with a zero cell count, see Details.
allincr	A logical indicating if incr is added to each cell frequency of all studies if at least one study has a zero cell count. If FALSE (default), incr is added only to each cell frequency of studies with a zero cell count.
addincr	A logical indicating if incr is added to each cell frequency of all studies irrespective of zero cell counts.
model.glmm	A character string indicating which GLMM should be used. One of "UM.FS", "UM.RS", and "CM.EL", see Details.
level	The level used to calculate confidence intervals for individual studies.
level.comb	The level used to calculate confidence intervals for pooled estimates.
comb.fixed	A logical indicating whether a fixed effect meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
prediction	A logical indicating whether a prediction interval should be printed.
level.predict	The level used to calculate prediction interval for a new study.
hakn	A logical indicating whether the method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
method.tau	A character string indicating which method is used to estimate the between-study variance τ^2. Either "DL", "PM", "REML", "ML", "HS", "SJ", "HE", or "EB", can be abbreviated.
tau.preset	Prespecified value for the square-root of the between-study variance τ^2.
TE.tau	Overall treatment effect used to estimate the between-study variance τ^2.
tau.common	A logical indicating whether tau-squared should be the same across subgroups.
method.bias	A character string indicating which test for funnel plot asymmetry is to be used. Either "linreg" or "rank", can be abbreviated. See function metabias
n.e	Number of observations in experimental group (optional).
n.c	Number of observations in control group (optional).
backtransf	A logical indicating whether results for incidence rate ratio (sm="IRR") should be back transformed in printouts and plots. If TRUE (default), results will be presented as incidence rate ratios; otherwise log incidence rate ratios will be shown.
irscale	A numeric defining a scaling factor for printing of incidence rate differences.
irunit	A character string specifying the time unit used to calculate rates, e.g. person-years.
title	Title of meta-analysis / systematic review.
complab	Comparison label.
outclab	Outcome label.
label.e	Label for experimental group.
label.c	Label for control group.

label.left	Graph label on left side of forest plot.
label.right	Graph label on right side of forest plot.
byvar	An optional vector containing grouping information (must be of same length as event.e).
bylab	A character string with a label for the grouping variable.
print.byvar	A logical indicating whether the name of the grouping variable should be printed in front of the group labels.
byseparator	A character string defining the separator between label and levels of grouping variable.
keepdata	A logical indicating whether original data (set) should be kept in meta object.
warn	A logical indicating whether warnings should be printed (e.g., if incr is added to studies with zero cell frequencies).
control	An optional list to control the iterative process to estimate the between-study variance tau^2. This argument is passed on to rma.uni or rma.glmm, respectively.
...	Additional arguments passed on to rma.glmm function.

Details

Treatment estimates and standard errors are calculated for each study. The following measures of treatment effect are available:

- Incidence Rate Ratio (sm="IRR")
- Incidence Rate Difference (sm="IRD")

For several arguments defaults settings are utilised (assignments using gs function). These defaults can be changed using the settings.meta function.

Internally, both fixed effect and random effects models are calculated regardless of values choosen for arguments comb.fixed and comb.random. Accordingly, the estimate for the random effects model can be extracted from component TE.random of an object of class "meta" even if argument comb.random=FALSE. However, all functions in R package **meta** will adequately consider the values for comb.fixed and comb.random. E.g. function print.meta will not print results for the random effects model if comb.random=FALSE.

By default, both fixed effect and random effects models are considered (see arguments comb.fixed and comb.random). If method is "MH" (default), the Mantel-Haenszel method is used to calculate the fixed effect estimate (Greenland & Robbins, 1985); if method is "Inverse", inverse variance weighting is used for pooling; if method is "Cochran", the Cochran method is used for pooling (Bayne-Jones, 1964, Chapter 8).

A distinctive and frequently overlooked advantage of incidence rates is that individual patient data (IPD) can be extracted from count data. Accordingly, statistical methods for IPD, i.e., generalised linear mixed models, can be utilised in a meta-analysis of incidence rate ratios (Stijnen et al., 2010). These methods are available (argument method = "GLMM") by calling the rma.glmm function from R package **metafor** internally. Three different GLMMs are available for meta-analysis of incidence rate ratios using argument model.glmm (which corresponds to argument model in the rma.glmm function):

- Poisson regression model with fixed study effects (default)

  (`model.glmm = "UM.FS"`, i.e., **U**nconditional **M**odel - **F**ixed **S**tudy effects)

- Mixed-effects Poisson regression model with random study effects

  (`model.glmm = "UM.RS"`, i.e., **U**nconditional **M**odel - **R**andom **S**tudy effects)

- Generalised linear mixed model (conditional Poisson-Normal)

  (`model.glmm = "CM.EL"`, i.e., **C**onditional **M**odel - **E**xact **L**ikelihood)

Details on these three GLMMs as well as additional arguments which can be provided using argument '`...`' in `metainc` are described in `rma.glmm` where you can also find information on the iterative algorithms used for estimation. Note, regardless of which value is used for argument `model.glmm`, results for two different GLMMs are calculated: fixed effect model (with fixed treatment effect) and random effects model (with random treatment effects).

For studies with a zero cell count, by default, 0.5 is added to all cell frequencies of these studies (argument `incr`). This continuity correction is used both to calculate individual study results with confidence limits and to conduct meta-analysis based on the inverse variance method. For Mantel-Haenszel method, Cochran method, and GLMMs, nothing is added to zero cell counts. Accordingly, estimates for these methods are not defined if the number of events is zero in all studies either in the experimental or control group.

Argument `byvar` can be used to conduct subgroup analysis for all methods but GLMMs. Instead use the `metareg` function for GLMMs which can also be used for continuous covariates.

A prediction interval for treatment effect of a new study is calculated (Higgins et al., 2009) if arguments `prediction` and `comb.random` are `TRUE`.

R function `update.meta` can be used to redo the meta-analysis of an existing metainc object by only specifying arguments which should be changed.

For the random effects, the method by Hartung and Knapp (2003) is used to adjust test statistics and confidence intervals if argument `hakn=TRUE`.

The DerSimonian-Laird estimate (1986) is used in the random effects model if `method.tau="DL"`. The iterative Paule-Mandel method (1982) to estimate the between-study variance is used if argument `method.tau="PM"`. Internally, R function `paulemandel` is called which is based on R function mpaule.default from R package **metRology** from S.L.R. Ellison <s.ellison at lgc.co.uk>.

If R package **metafor** (Viechtbauer 2010) is installed, the following methods to estimate the between-study variance τ^2 (argument `method.tau`) are also available:

- Restricted maximum-likelihood estimator (`method.tau="REML"`)
- Maximum-likelihood estimator (`method.tau="ML"`)
- Hunter-Schmidt estimator (`method.tau="HS"`)
- Sidik-Jonkman estimator (`method.tau="SJ"`)
- Hedges estimator (`method.tau="HE"`)
- Empirical Bayes estimator (`method.tau="EB"`).

For these methods the R function `rma.uni` of R package **metafor** is called internally. See help page of R function `rma.uni` for more details on these methods to estimate between-study variance.

Value

An object of class `c("metainc", "meta")` with corresponding `print`, `summary`, and `forest` functions. The object is a list containing the following components:

`event.e, time.e, event.c, time.c, studlab, exclude,`

`sm, method, incr, allincr, addincr, model.glmm, warn,`

`level, level.comb, comb.fixed, comb.random,`

`hakn, method.tau, tau.preset, TE.tau, method.bias,`

`tau.common, title, complab, outclab,`

`label.e, label.c, label.left, label.right,`

`byvar, bylab, print.byvar, byseparator`
: As defined above.

`TE, seTE`
: Estimated treatment effect and standard error of individual studies.

`lower, upper`
: Lower and upper confidence interval limits for individual studies.

`zval, pval`
: z-value and p-value for test of treatment effect for individual studies.

`w.fixed, w.random`
: Weight of individual studies (in fixed and random effects model).

`TE.fixed, seTE.fixed`
: Estimated overall treatment effect and standard error (fixed effect model).

`lower.fixed, upper.fixed`
: Lower and upper confidence interval limits (fixed effect model).

`zval.fixed, pval.fixed`
: z-value and p-value for test of overall treatment effect (fixed effect model).

`TE.random, seTE.random`
: Estimated overall treatment effect and standard error (random effects model).

`lower.random, upper.random`
: Lower and upper confidence interval limits (random effects model).

`zval.random, pval.random`
: z-value or t-value and corresponding p-value for test of overall treatment effect (random effects model).

`prediction, level.predict`
: As defined above.

`seTE.predict`
: Standard error utilised for prediction interval.

`lower.predict, upper.predict`
: Lower and upper limits of prediction interval.

`k`
: Number of studies combined in meta-analysis.

`Q`
: Heterogeneity statistic Q.

`df.Q`
: Degrees of freedom for heterogeneity statistic.

`pval.Q` P-value of heterogeneity test.

`Q.LRT` Heterogeneity statistic for likelihood-ratio test (only if `method = "GLMM"`).

`df.Q.LRT` Degrees of freedom for likelihood-ratio test

`pval.Q.LRT` P-value of likelihood-ratio test.

`tau` Square-root of between-study variance.

`se.tau` Standard error of square-root of between-study variance.

`C` Scaling factor utilised internally to calculate common tau-squared across subgroups.

`sparse` Logical flag indicating if any study included in meta-analysis has any zero cell frequencies.

`incr.event` Increment added to number of events.

`df.hakn` Degrees of freedom for test of treatment effect for Hartung-Knapp method (only if `hakn=TRUE`).

`k.MH` Number of studies combined in meta-analysis using Mantel-Haenszel method.

`bylevs` Levels of grouping variable - if `byvar` is not missing.

`TE.fixed.w, seTE.fixed.w`
Estimated treatment effect and standard error in subgroups (fixed effect model) - if `byvar` is not missing.

`lower.fixed.w, upper.fixed.w`
Lower and upper confidence interval limits in subgroups (fixed effect model) - if `byvar` is not missing.

`zval.fixed.w, pval.fixed.w`
z-value and p-value for test of treatment effect in subgroups (fixed effect model) - if `byvar` is not missing.

`TE.random.w, seTE.random.w`
Estimated treatment effect and standard error in subgroups (random effects model) - if `byvar` is not missing.

`lower.random.w, upper.random.w`
Lower and upper confidence interval limits in subgroups (random effects model) - if `byvar` is not missing.

`zval.random.w, pval.random.w`
z-value or t-value and corresponding p-value for test of treatment effect in subgroups (random effects model) - if `byvar` is not missing.

`w.fixed.w, w.random.w`
Weight of subgroups (in fixed and random effects model) - if `byvar` is not missing.

`df.hakn.w` Degrees of freedom for test of treatment effect for Hartung-Knapp method in subgroups - if `byvar` is not missing and `hakn=TRUE`.

`n.harmonic.mean.w`
Harmonic mean of number of observations in subgroups (for back transformation of Freeman-Tukey Double arcsine transformation) - if `byvar` is not missing.

`event.e.w` Number of events in experimental group in subgroups - if `byvar` is not missing.

`time.e.w` Total person time in subgroups (experimental group) - if `byvar` is not missing.

`n.e.w` Number of observations in experimental group in subgroups - if `byvar` is not missing.

`event.c.w` Number of events in control group in subgroups - if `byvar` is not missing.

`time.c.w` Total person time in subgroups (control group) - if `byvar` is not missing.

`n.c.w` Number of observations in control group in subgroups - if `byvar` is not missing.

`k.w` Number of studies combined within subgroups - if `byvar` is not missing.

`k.all.w` Number of all studies in subgroups - if `byvar` is not missing.

`Q.w.fixed` Overall within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.w.random` Overall within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing (only calculated if argument `tau.common` is TRUE).

`df.Q.w` Degrees of freedom for test of overall within subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.w.fixed` P-value of within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.w.random` P-value of within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`Q.b.fixed` Overall between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.b.random` Overall between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`df.Q.b` Degrees of freedom for test of overall between subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.b.fixed` P-value of between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.b.random` P-value of between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`tau.w` Square-root of between-study variance within subgroups - if `byvar` is not missing.

`C.w` Scaling factor utilised internally to calculate common tau-squared across subgroups - if `byvar` is not missing.

`H.w` Heterogeneity statistic H within subgroups - if `byvar` is not missing.

`lower.H.w, upper.H.w` Lower and upper confidence limti for heterogeneity statistic H within subgroups - if `byvar` is not missing.

`I2.w` Heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`lower.I2.w, upper.I2.w` Lower and upper confidence limti for heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

keepdata	As defined above.
data	Original data (set) used in function call (if keepdata=TRUE).
subset	Information on subset of original data used in meta-analysis (if keepdata=TRUE).
.glmm.fixed	GLMM object generated by call of rma.glmm function (fixed effect model).
.glmm.random	GLMM object generated by call of rma.glmm function (random effects model).
call	Function call.
version	Version of R package **meta** used to create object.
version.metafor	Version of R package **metafor** used for GLMMs.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Bayne-Jones S et al. (1964), Smoking and Health: Report of the Advisory Committee to the Surgeon General of the United States. U-23 Department of Health, Education, and Welfare. Public Health Service Publication No. 1103. http://profiles.nlm.nih.gov/ps/retrieve/ResourceMetadata/NNBBMQ

DerSimonian R & Laird N (1986), Meta-analysis in clinical trials. *Controlled Clinical Trials*, **7**, 177–188.

Greenland S & Robins JM (1985), Estimation of a common effect parameter from sparse follow-up data. *Biometrics*, **41**, 55–68.

Hartung J & Knapp G (2001), A Refined Method for the Meta-analysis of Controlled Clinical Trials with Binary Outcome. *Statistics in Medicine*, **20**, 3875–89.

Higgins JPT, Thompson SG, Spiegelhalter DJ (2009), A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society: Series A*, **172**, 137–159.

Knapp G & Hartung J (2003), Improved Tests for a Random Effects Meta-regression with a Single Covariate. *Statistics in Medicine*, **22**, 2693–710, doi: 10.1002/sim.1482 .

Paule RC & Mandel J (1982), Consensus values and weighting factors. *Journal of Research of the National Bureau of Standards*, **87**, 377–385.

Stijnen T, Hamza TH, 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**, 3046–67.

Viechtbauer W (2010), Conducting Meta-Analyses in R with the Metafor Package. *Journal of Statistical Software*, **36**, 1–48.

See Also

metabin, update.meta, print.meta

Examples

```
data(smoking)

m1 <- metainc(d.smokers, py.smokers,
              d.nonsmokers, py.nonsmokers,
              data=smoking, studlab=study)
print(m1, digits=2)

m2 <- metainc(d.smokers, py.smokers,
              d.nonsmokers, py.nonsmokers,
              data=smoking, studlab=study,
              method="Cochran")
print(m2, digits=2)

data(lungcancer)

m3 <- metainc(d.smokers, py.smokers,
              d.nonsmokers, py.nonsmokers,
              data=lungcancer, studlab=study)
print(m3, digits=2)


# Redo Cochran meta-analysis with inflated standard errors
#
# All cause mortality
#
TEa <- log( (smoking$d.smokers/smoking$py.smokers) /
            (smoking$d.nonsmokers/smoking$py.nonsmokers)
          )
seTEa <- sqrt(1/smoking$d.smokers +
              1/smoking$d.nonsmokers + 2.5/smoking$d.nonsmokers)
#
metagen(TEa, seTEa, sm="IRR", studlab=smoking$study)

# Lung cancer mortality
#
TEl <- log( (lungcancer$d.smokers/lungcancer$py.smokers) /
            (lungcancer$d.nonsmokers/lungcancer$py.nonsmokers)
          )
seTEl <- sqrt(1/lungcancer$d.smokers +
              1/lungcancer$d.nonsmokers + 2.25/lungcancer$d.nonsmokers)
#
metagen(TEl, seTEl, sm="IRR", studlab=lungcancer$study)


## Not run:
#
# Meta-analysis using generalised linear mixed models
# (only if R packages 'metafor' and 'lme4' are available)
#

#
```

```
# Poisson regression model (fixed study effects)
#
m4 <- metainc(d.smokers, py.smokers, d.nonsmokers, py.nonsmokers,
              data = smoking, studlab = study, method = "GLMM")
m4
#
# Mixed-effects Poisson regression model (random study effects)
#
update(m4, model.glmm = "UM.RS", nAGQ = 1)
#
# Generalised linear mixed model (conditional Poisson-Normal)
#
update(m4, model.glmm = "CM.EL")

## End(Not run)
```

`metainf` *Influence analysis in meta-analysis using leave-one-out method*

Description

Performs an influence analysis. Pooled estimates are calculated omitting one study at a time.

Usage

```
metainf(x, pooled, sortvar)
```

Arguments

`x`	An object of class `meta`.
`pooled`	A character string indicating whether a fixed effect or random effects model is used for pooling. Either missing (see Details), `"fixed"` or `"random"`, can be abbreviated.
`sortvar`	An optional vector used to sort the individual studies (must be of same length as `x$TE`).

Details

Performs a influence analysis; pooled estimates are calculated omitting one study at a time. Studies are sorted according to `sortvar`.

Information from object `x` is utilised if argument `pooled` is missing. A fixed effect model is assumed (`pooled="fixed"`) if argument `x$comb.fixed` is `TRUE`; a random effects model is assumed (`pooled="random"`) if argument `x$comb.random` is `TRUE` and `x$comb.fixed` is `FALSE`.

Value

An object of class `c("metainf", "meta")` with corresponding `print`, and `forest` functions. The object is a list containing the following components:

`TE, seTE`	Estimated treatment effect and standard error of pooled estimate in influence analysis.
`lower, upper`	Lower and upper confidence interval limits.
`studlab`	Study label describing omission of studies.
`p.value`	P-value for test of overall effect.
`w`	Sum of weights from fixed effect or random effects model.
`I2`	Heterogeneity statistic I2.
`Rb`	Heterogeneity statistic Rb.
`tau`	Square-root of between-study variance.
`df.hakn`	Degrees of freedom for test of treatment effect for Hartung-Knapp method (only if `hakn=TRUE`).
`sm`	Summary measure.
`method`	Method used for pooling.
`k`	Number of studies combined in meta-analysis.
`pooled`	As defined above.
`comb.fixed`	A logical indicating whether analysis is based on fixed effect model.
`comb.random`	A logical indicating whether analysis is based on random effects model.
`TE.fixed, seTE.fixed`	Value is `NA`.
`TE.random, seTE.random`	Value is `NA`.
`Q`	Value is `NA`.
`level.comb`	The level used to calculate confidence intervals for pooled estimates.
`hakn`	A logical indicating whether the method by Hartung and Knapp is used to adjust test statistics and confidence intervals.
`method.tau`	A character string indicating which method is used to estimate the between-study variance τ^2.
`tau.preset`	Prespecified value for the square-root of the between-study variance τ^2.
`TE.tau`	Overall treatment effect used to estimate the between-study variance τ^2.
`n.harmonic.mean`	Harmonic mean of number of observations (for back transformation of Freeman-Tukey Double arcsine transformation).
`version`	Version of R package **meta** used to create object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Cooper H & Hedges LV (1994), *The Handbook of Research Synthesis*. Newbury Park, CA: Russell Sage Foundation.

See Also

metabin, metacont, print.meta

Examples

```
data(Fleiss93)
meta1 <- metabin(event.e, n.e, event.c, n.c,
                 data=Fleiss93, studlab=study,
                 sm="RR", method="I")
meta1

metainf(meta1)
metainf(meta1, pooled="random")

forest(metainf(meta1))
forest(metainf(meta1), layout="revman5")
forest(metainf(meta1, pooled="random"))

metainf(meta1, sortvar=study)
metainf(meta1, sortvar=7:1)

meta2 <- update(meta1, title="Fleiss93 meta-analysis",
                backtransf=FALSE)
metainf(meta2)


data(Fleiss93cont)
meta3 <- metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c,
                  data = Fleiss93cont, sm = "SMD")
metainf(meta3)
```

---

metamean *Meta-analysis of single means*

Description

Calculation of an overall mean from studies reporting a single mean using the inverse varinace method for pooling; inverse variance weighting is used for pooling.

Usage

```
metamean(n, mean, sd, studlab,
         data=NULL, subset=NULL, exclude=NULL,
         sm=gs("smmean"),
```

```
level=gs("level"), level.comb=gs("level.comb"),
comb.fixed=gs("comb.fixed"), comb.random=gs("comb.random"),
hakn=gs("hakn"),
method.tau=gs("method.tau"), tau.preset=NULL, TE.tau=NULL,
tau.common=gs("tau.common"),
prediction=gs("prediction"), level.predict=gs("level.predict"),
null.effect=NA,
method.bias=gs("method.bias"),
backtransf=gs("backtransf"),
title=gs("title"), complab=gs("complab"), outclab="",
byvar, bylab, print.byvar=gs("print.byvar"),
byseparator=gs("byseparator"),
keepdata=gs("keepdata"),
warn=gs("warn"),
control=NULL)
```

Arguments

n	Number of observations.
mean	Estimated mean.
sd	Standard deviation.
studlab	An optional vector with study labels.
data	An optional data frame containing the study information.
subset	An optional vector specifying a subset of studies to be used.
exclude	An optional vector specifying studies to exclude from meta-analysis, however, to include in printouts and forest plots.
level	The level used to calculate confidence intervals for individual studies.
level.comb	The level used to calculate confidence intervals for pooled estimates.
comb.fixed	A logical indicating whether a fixed effect meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
prediction	A logical indicating whether a prediction interval should be printed.
level.predict	The level used to calculate prediction interval for a new study.
null.effect	A numeric value specifying the effect under the null hypothesis.
hakn	A logical indicating whether the method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
method.tau	A character string indicating which method is used to estimate the between-study variance τ^2. Either "DL", "PM", "REML", "ML", "HS", "SJ", "HE", or "EB", can be abbreviated.
tau.preset	Prespecified value for the square-root of the between-study variance τ^2.
TE.tau	Overall treatment effect used to estimate the between-study variance tau-squared.
tau.common	A logical indicating whether tau-squared should be the same across subgroups.

method.bias	A character string indicating which test is to be used. Either "rank", "linreg", or "mm", can be abbreviated. See function metabias
backtransf	A logical indicating whether results should be back transformed in printouts and plots for sm="MLN". If TRUE (default), results will be presented as means; otherwise logarithm of means will be shown.
title	Title of meta-analysis / systematic review.
complab	Comparison label.
outclab	Outcome label.
sm	A character string indicating which summary measure ("MRAW" or "MLN") is to be used for pooling of studies.
byvar	An optional vector containing grouping information (must be of same length as n).
bylab	A character string with a label for the grouping variable.
print.byvar	A logical indicating whether the name of the grouping variable should be printed in front of the group labels.
byseparator	A character string defining the separator between label and levels of grouping variable.
keepdata	A logical indicating whether original data (set) should be kept in meta object.
warn	A logical indicating whether warnings should be printed (e.g., if studies are excluded from meta-analysis due to zero standard deviations).
control	An optional list to control the iterative process to estimate the between-study variance tau^2. This argument is passed on to rma.uni.

Details

Fixed effect and random effects meta-analysis of single means to calculate an overall mean; inverse variance weighting is used for pooling. The following transformations of means are implemented to calculate an overall mean:

- Raw, i.e. untransformed, means (sm="MRAW", default)
- Log transformed means (sm="MLN")

Note, you should use R function metacont to compare means of pairwise comparisons instead of using metamean for each treatment arm separately which will break randomisation in randomised controlled trials.

Calculations are conducted on the log scale if sm="ROM". Accordingly, list elements TE, TE.fixed, and TE.random contain the logarithm of means. In printouts and plots these values are back transformed if argument backtransf=TRUE.

For several arguments defaults settings are utilised (assignments using gs function). These defaults can be changed using the settings.meta function.

Internally, both fixed effect and random effects models are calculated regardless of values choosen for arguments comb.fixed and comb.random. Accordingly, the estimate for the random effects model can be extracted from component TE.random of an object of class "meta" even if argument comb.random=FALSE. However, all functions in R package **meta** will adequately consider the values

for comb.fixed and comb.random. E.g. function print.meta will not print results for the random effects model if comb.random=FALSE.

The function metagen is called internally to calculate individual and overall treatment estimates and standard errors.

A prediction interval for treatment effect of a new study is calculated (Higgins et al., 2009) if arguments prediction and comb.random are TRUE.

R function update.meta can be used to redo the meta-analysis of an existing metamean object by only specifying arguments which should be changed.

For the random effects, the method by Hartung and Knapp (2001) / Knapp and Hartung (2003) is used to adjust test statistics and confidence intervals if argument hakn=TRUE.

The DerSimonian-Laird estimate (1986) is used in the random effects model if method.tau="DL". The iterative Paule-Mandel method (1982) to estimate the between-study variance is used if argument method.tau="PM". Internally, R function paulemandel is called which is based on R function mpaule.default from R package **metRology** from S.L.R. Ellison <s.ellison at lgc.co.uk>.

If R package **metafor** (Viechtbauer 2010) is installed, the following methods to estimate the between-study variance τ^2 (argument method.tau) are also available:

- Restricted maximum-likelihood estimator (method.tau="REML")
- Maximum-likelihood estimator (method.tau="ML")
- Hunter-Schmidt estimator (method.tau="HS")
- Sidik-Jonkman estimator (method.tau="SJ")
- Hedges estimator (method.tau="HE")
- Empirical Bayes estimator (method.tau="EB").

For these methods the R function rma.uni of R package **metafor** is called internally. See help page of R function rma.uni for more details on these methods to estimate between-study variance.

Value

An object of class c("metamean", "meta") with corresponding print, summary, and forest functions. The object is a list containing the following components:

n, mean, sd,
studlab, exclude, sm, level, level.comb,

comb.fixed, comb.random,

hakn, method.tau, tau.preset, TE.tau, method.bias,

tau.common, title, complab, outclab,

byvar, bylab, print.byvar, byseparator, warn
: As defined above.

TE, seTE
: Estimated effect (mean or log mean) and standard error of individual studies.

lower, upper
: Lower and upper confidence interval limits for individual studies.

`zval, pval` z-value and p-value for test of overall effect for individual studies.

`w.fixed, w.random`
Weight of individual studies (in fixed and random effects model).

`TE.fixed, seTE.fixed`
Estimated overall effect (mean or log mean) and standard error (fixed effect model).

`lower.fixed, upper.fixed`
Lower and upper confidence interval limits (fixed effect model).

`zval.fixed, pval.fixed`
z-value and p-value for test of overall effect (fixed effect model).

`TE.random, seTE.random`
Estimated overall effect (mean or log mean) and standard error (random effects model).

`lower.random, upper.random`
Lower and upper confidence interval limits (random effects model).

`zval.random, pval.random`
z-value or t-value and corresponding p-value for test of overall effect (random effects model).

`prediction, level.predict`
As defined above.

`seTE.predict` Standard error utilised for prediction interval.

`lower.predict, upper.predict`
Lower and upper limits of prediction interval.

`k` Number of studies combined in meta-analysis.

`Q` Heterogeneity statistic.

`tau` Square-root of between-study variance.

`se.tau` Standard error of square-root of between-study variance.

`C` Scaling factor utilised internally to calculate common tau-squared across subgroups.

`method` Pooling method: `"Inverse"`.

`df.hakn` Degrees of freedom for test of treatment effect for Hartung-Knapp method (only if `hakn=TRUE`).

`bylevs` Levels of grouping variable - if `byvar` is not missing.

`TE.fixed.w, seTE.fixed.w`
Estimated effect and standard error in subgroups (fixed effect model) - if `byvar` is not missing.

`lower.fixed.w, upper.fixed.w`
Lower and upper confidence interval limits in subgroups (fixed effect model) - if `byvar` is not missing.

`zval.fixed.w, pval.fixed.w`
z-value and p-value for test of treatment effect in subgroups (fixed effect model) - if `byvar` is not missing.

`TE.random.w, seTE.random.w`
: Estimated effect and standard error in subgroups (random effects model) - if `byvar` is not missing.

`lower.random.w, upper.random.w`
: Lower and upper confidence interval limits in subgroups (random effects model) - if `byvar` is not missing.

`zval.random.w, pval.random.w`
: z-value or t-value and corresponding p-value for test of effect in subgroups (random effects model) - if `byvar` is not missing.

`w.fixed.w, w.random.w`
: Weight of subgroups (in fixed and random effects model) - if `byvar` is not missing.

`df.hakn.w`
: Degrees of freedom for test of effect for Hartung-Knapp method in subgroups - if `byvar` is not missing and `hakn=TRUE`.

`n.e.w`
: Number of observations in experimental group in subgroups - if `byvar` is not missing.

`n.c.w`
: Number of observations in control group in subgroups - if `byvar` is not missing.

`k.w`
: Number of studies combined within subgroups - if `byvar` is not missing.

`k.all.w`
: Number of all studies in subgroups - if `byvar` is not missing.

`Q.w.fixed`
: Overall within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.w.random`
: Overall within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing (only calculated if argument `tau.common` is TRUE).

`df.Q.w`
: Degrees of freedom for test of overall within subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.w.fixed`
: P-value of within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.w.random`
: P-value of within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`Q.b.fixed`
: Overall between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.b.random`
: Overall between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`df.Q.b`
: Degrees of freedom for test of overall between subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.b.fixed`
: P-value of between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.b.random`
: P-value of between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`tau.w`
: Square-root of between-study variance within subgroups - if `byvar` is not missing.

`C.w` Scaling factor utilised internally to calculate common tau-squared across subgroups - if `byvar` is not missing.

`H.w` Heterogeneity statistic H within subgroups - if `byvar` is not missing.

`lower.H.w, upper.H.w`
Lower and upper confidence limti for heterogeneity statistic H within subgroups - if `byvar` is not missing.

`I2.w` Heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`lower.I2.w, upper.I2.w`
Lower and upper confidence limti for heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`keepdata` As defined above.

`data` Original data (set) used in function call (if `keepdata=TRUE`).

`subset` Information on subset of original data used in meta-analysis (if `keepdata=TRUE`).

`call` Function call.

`version` Version of R package **meta** used to create object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

DerSimonian R & Laird N (1986), Meta-analysis in clinical trials. *Controlled Clinical Trials*, **7**, 177–88.

Hartung J & Knapp G (2001), On tests of the overall treatment effect in meta-analysis with normally distributed responses. *Statistics in Medicine*, **20**, 1771–82. doi: 10.1002/sim.791 .

Higgins JPT, Thompson SG, Spiegelhalter DJ (2009), A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society: Series A*, **172**, 137–59.

Knapp G & Hartung J (2003), Improved Tests for a Random Effects Meta-regression with a Single Covariate. *Statistics in Medicine*, **22**, 2693–710, doi: 10.1002/sim.1482 .

Paule RC & Mandel J (1982), Consensus values and weighting factors. *Journal of Research of the National Bureau of Standards*, **87**, 377–85.

Viechtbauer W (2010), Conducting Meta-Analyses in R with the Metafor Package. *Journal of Statistical Software*, **36**, 1–48.

See Also

`update.meta`, `metamean`, `metagen`

Examples

```
m1 <- metamean(rep(100, 3), 1:3, rep(1, 3))
m2 <- update(m1, sm="MLN")

m1
m2
```

```
# With test for overall mean equal to 2
#
update(m1, null.effect=2)
update(m2, null.effect=2)

# Print results without back-transformation
#
print(m1, backtransf=FALSE)
update(m2, backtransf=FALSE)
update(m1, null.effect=2, backtransf=FALSE)
update(m2, null.effect=2, backtransf=FALSE)
```

---

metaprop *Meta-analysis of single proportions*

Description

Calculation of an overall proportion from studies reporting a single proportion. Inverse variance method and generalised linear mixed model (GLMM) are available for pooling. For GLMMs, the rma.glmm function from R package **metafor** (Viechtbauer 2010) is called internally.

Usage

```
metaprop(event, n, studlab,
         data=NULL, subset=NULL, exclude=NULL,
         method = "Inverse",
         sm=gs("smprop"),
         incr=gs("incr"), allincr=gs("allincr"),
         addincr=gs("addincr"),
         method.ci=gs("method.ci"),
         level=gs("level"), level.comb=gs("level.comb"),
         comb.fixed=gs("comb.fixed"), comb.random=gs("comb.random"),
         hakn=gs("hakn"),
         method.tau=
         ifelse(!is.na(charmatch(tolower(method), "glmm", nomatch = NA)),
                "ML", gs("method.tau")),
         tau.preset=NULL, TE.tau=NULL,
         tau.common=gs("tau.common"),
         prediction=gs("prediction"), level.predict=gs("level.predict"),
         null.effect=NA,
         method.bias=gs("method.bias"),
         backtransf=gs("backtransf"),
         pscale=1,
         title=gs("title"), complab=gs("complab"), outclab="",
         byvar, bylab, print.byvar=gs("print.byvar"),
         byseparator = gs("byseparator"),
         keepdata=gs("keepdata"),
```

```
          warn=gs("warn"),
          control=NULL,
          ...)
```

Arguments

event	Number of events.
n	Number of observations.
studlab	An optional vector with study labels.
data	An optional data frame containing the study information, i.e., event and n.
subset	An optional vector specifying a subset of studies to be used.
exclude	An optional vector specifying studies to exclude from meta-analysis, however, to include in printouts and forest plots.
method	A character string indicating which method is to be used for pooling of studies. One of "Inverse" and "GLMM", can be abbreviated.
sm	A character string indicating which summary measure ("PFT", "PAS", "PRAW", "PLN", or "PLOGIT") is to be used for pooling of studies, see Details.
incr	A numeric which is added to event number and sample size of studies with zero or all events, i.e., studies with an event probability of either 0 or 1.
allincr	A logical indicating if incr is considered for all studies if at least one study has either zero or all events. If FALSE (default), incr is considered only in studies with zero or all events.
addincr	A logical indicating if incr is used for all studies irrespective of number of events.
method.ci	A character string indicating which method is used to calculate confidence intervals for individual studies, see Details.
level	The level used to calculate confidence intervals for individual studies.
level.comb	The level used to calculate confidence intervals for pooled estimates.
comb.fixed	A logical indicating whether a fixed effect meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
prediction	A logical indicating whether a prediction interval should be printed.
level.predict	The level used to calculate prediction interval for a new study.
hakn	A logical indicating whether the method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
method.tau	A character string indicating which method is used to estimate the between-study variance τ^2, see Details.
tau.preset	Prespecified value for the square-root of the between-study variance τ^2.
TE.tau	Overall treatment effect used to estimate the between-study variance tau-squared.
tau.common	A logical indicating whether tau-squared should be the same across subgroups.
null.effect	A numeric value specifying the effect under the null hypothesis.

method.bias	A character string indicating which test is to be used. Either "rank", "linreg", or "mm", can be abbreviated. See function metabias.
backtransf	A logical indicating whether results for transformed proportions (argument sm!="PRAW") should be back transformed in printouts and plots. If TRUE (default), results will be presented as proportions; otherwise transformed proportions will be shown. See Details for presentation of confidence intervals.
pscale	A numeric defining a scaling factor for printing of single event probabilities.
title	Title of meta-analysis / systematic review.
complab	Comparison label.
outclab	Outcome label.
byvar	An optional vector containing grouping information (must be of same length as event).
bylab	A character string with a label for the grouping variable.
print.byvar	A logical indicating whether the name of the grouping variable should be printed in front of the group labels.
byseparator	A character string defining the separator between label and levels of grouping variable.
keepdata	A logical indicating whether original data (set) should be kept in meta object.
warn	A logical indicating whether the addition of incr to studies with zero or all events should result in a warning.
control	An optional list to control the iterative process to estimate the between-study variance tau^2. This argument is passed on to rma.uni or rma.glmm, respectively.
...	Additional arguments passed on to rma.glmm function.

Details

Fixed effect and random effects meta-analysis of single proportions to calculate an overall proportion. The following transformations of proportions are implemented to calculate an overall proportion:

- Logit transformation (sm="PLOGIT", default)
- Log transformation (sm="PLN")
- Freeman-Tukey Double arcsine transformation (sm="PFT")
- Arcsine transformation (sm="PAS")
- Raw, i.e. untransformed, proportions (sm="PRAW")

Note, you should use R function metabin to compare proportions of pairwise comparisons instead of using metaprop for each treatment arm separately which will break randomisation in randomised controlled trials.

Various methods are available to calculate confidence intervals for individual study results (see Agresti & Coull 1998; Newcombe 1988):

- Clopper-Pearson interval also called 'exact' binomial interval (method.ci="CP", default)

- Wilson Score interval (method.ci="WS")
- Wilson Score interval with continuity correction (method.ci="WSCC")
- Agresti-Coull interval (method.ci="AC")
- Simple approximation interval (method.ci="SA")
- Simple approximation interval with continuity correction (method.ci="SACC")
- Normal approximation interval based on summary measure, i.e. defined by argument sm (method.ci="NAsm")

Note, with exception of the normal approximation based on the summary measure, i.e. method.ci="NAsm", the same confidence interval is calculated for any summary measure (argument sm) as only number of events and observations are used in the calculation disregarding the chosen summary measure. Results will be presented for transformed proportions if argument backtransf=FALSE in the print.meta, print.summary.meta, or forest.meta function. In this case, argument method.ci="NAsm" is used, i.e. confidence intervals based on the normal approximation based on the summary measure.

Argument pscale can be used to rescale proportions, e.g. pscale=1000 means that proportions are expressed as events per 1000 observations. This is useful in situations with (very) low event probabilities.

For several arguments defaults settings are utilised (assignments using gs function). These defaults can be changed using the settings.meta function.

Internally, both fixed effect and random effects models are calculated regardless of values choosen for arguments comb.fixed and comb.random. Accordingly, the estimate for the random effects model can be extracted from component TE.random of an object of class "meta" even if argument comb.random=FALSE. However, all functions in R package **meta** will adequately consider the values for comb.fixed and comb.random. E.g. function print.meta will not print results for the random effects model if comb.random=FALSE.

A distinctive and frequently overlooked advantage of binary data is that individual patient data (IPD) can be extracted. Accordingly, a random intercept logistic regression model can be utilised for the meta-analysis of proportions (Stijnen et al., 2010). This method is available (argument method = "GLMM") by calling the rma.glmm function from R package **metafor** internally.

If the summary measure is equal to "PRAW", "PLN", or "PLOGIT", a continuity correction is applied if any study has either zero or all events, i.e., an event probability of either 0 or 1. By default, 0.5 is used as continuity correction (argument incr). This continuity correction is used both to calculate individual study results with confidence limits and to conduct meta-analysis based on the inverse variance method. For GLMMs no continuity correction is used.

Argument byvar can be used to conduct subgroup analysis for all methods but GLMMs. Instead use the metareg function for GLMMs which can also be used for continuous covariates.

A prediction interval for treatment effect of a new study is calculated (Higgins et al., 2009) if arguments prediction and comb.random are TRUE.

R function update.meta can be used to redo the meta-analysis of an existing metaprop object by only specifying arguments which should be changed.

For the random effects, the method by Hartung and Knapp (2003) is used to adjust test statistics and confidence intervals if argument hakn=TRUE.

The DerSimonian-Laird estimate (1986) is used in the random effects model if `method.tau="DL"`. The iterative Paule-Mandel method (1982) to estimate the between-study variance is used if argument `method.tau="PM"`. Internally, R function `paulemandel` is called which is based on R function mpaule.default from R package **metRology** from S.L.R. Ellison <s.ellison at lgc.co.uk>.

If R package **metafor** (Viechtbauer 2010) is installed, the following methods to estimate the between-study variance τ^2 (argument `method.tau`) are also available:

- Restricted maximum-likelihood estimator (`method.tau="REML"`)
- Maximum-likelihood estimator (`method.tau="ML"`)
- Hunter-Schmidt estimator (`method.tau="HS"`)
- Sidik-Jonkman estimator (`method.tau="SJ"`)
- Hedges estimator (`method.tau="HE"`)
- Empirical Bayes estimator (`method.tau="EB"`).

For these methods the R function `rma.uni` of R package **metafor** is called internally. See help page of R function `rma.uni` for more details on these methods to estimate between-study variance.

Value

An object of class `c("metaprop", "meta")` with corresponding `print`, `summary`, and `forest` functions. The object is a list containing the following components:

`event, n, studlab, exclude,`

`sm, incr, allincr, addincr, method.ci,`

`level, level.comb,`
: As defined above.

`comb.fixed, comb.random,`

`hakn, method.tau, tau.preset, TE.tau, null.hypothesis,`

`method.bias, tau.common, title, complab, outclab,`

`byvar, bylab, print.byvar, byseparator, warn`

`TE, seTE`
: Estimated (un)transformed proportion and its standard error for individual studies.

`lower, upper`
: Lower and upper confidence interval limits for individual studies.

`zval, pval`
: z-value and p-value for test of treatment effect for individual studies.

`w.fixed, w.random`
: Weight of individual studies (in fixed and random effects model).

`TE.fixed, seTE.fixed`
: Estimated overall (un)transformed proportion and standard error (fixed effect model).

`lower.fixed, upper.fixed`
: Lower and upper confidence interval limits (fixed effect model).

`zval.fixed, pval.fixed`
: z-value and p-value for test of overall effect (fixed effect model).

`TE.random, seTE.random`
: Estimated overall (un)transformed proportion and standard error (random effects model).

`lower.random, upper.random`
: Lower and upper confidence interval limits (random effects model).

`zval.random, pval.random`
: z-value or t-value and corresponding p-value for test of overall effect (random effects model).

`prediction, level.predict`
: As defined above.

`seTE.predict`
: Standard error utilised for prediction interval.

`lower.predict, upper.predict`
: Lower and upper limits of prediction interval.

`k`
: Number of studies combined in meta-analysis.

`Q`
: Heterogeneity statistic Q.

`df.Q`
: Degrees of freedom for heterogeneity statistic.

`pval.Q`
: P-value of heterogeneity test.

`Q.LRT`
: Heterogeneity statistic for likelihood-ratio test (only if `method = "GLMM"`).

`df.Q.LRT`
: Degrees of freedom for likelihood-ratio test

`pval.Q.LRT`
: P-value of likelihood-ratio test.

`tau`
: Square-root of between-study variance.

`se.tau`
: Standard error of square-root of between-study variance.

`C`
: Scaling factor utilised internally to calculate common tau-squared across subgroups.

`method`
: A character string indicating method used for pooling: `"Inverse"`

`df.hakn`
: Degrees of freedom for test of treatment effect for Hartung-Knapp method (only if `hakn=TRUE`).

`bylevs`
: Levels of grouping variable - if `byvar` is not missing.

`TE.fixed.w, seTE.fixed.w`
: Estimated treatment effect and standard error in subgroups (fixed effect model) - if `byvar` is not missing.

`lower.fixed.w, upper.fixed.w`
: Lower and upper confidence interval limits in subgroups (fixed effect model) - if `byvar` is not missing.

`zval.fixed.w, pval.fixed.w`
: z-value and p-value for test of treatment effect in subgroups (fixed effect model) - if `byvar` is not missing.

`TE.random.w, seTE.random.w`
: Estimated treatment effect and standard error in subgroups (random effects model) - if `byvar` is not missing.

`lower.random.w, upper.random.w`
: Lower and upper confidence interval limits in subgroups (random effects model) - if `byvar` is not missing.

`zval.random.w, pval.random.w`
: z-value or t-value and corresponding p-value for test of treatment effect in subgroups (random effects model) - if `byvar` is not missing.

`w.fixed.w, w.random.w`
: Weight of subgroups (in fixed and random effects model) - if `byvar` is not missing.

`df.hakn.w`
: Degrees of freedom for test of treatment effect for Hartung-Knapp method in subgroups - if `byvar` is not missing and `hakn=TRUE`.

`n.harmonic.mean.w`
: Harmonic mean of number of observations in subgroups (for back transformation of Freeman-Tukey Double arcsine transformation) - if `byvar` is not missing.

`event.w`
: Number of events in subgroups - if `byvar` is not missing.

`n.w`
: Number of observations in subgroups - if `byvar` is not missing.

`k.w`
: Number of studies combined within subgroups - if `byvar` is not missing.

`k.all.w`
: Number of all studies in subgroups - if `byvar` is not missing.

`Q.w.fixed`
: Overall within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.w.random`
: Overall within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing (only calculated if argument `tau.common` is TRUE).

`df.Q.w`
: Degrees of freedom for test of overall within subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.w.fixed`
: P-value of within subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.w.random`
: P-value of within subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`Q.b.fixed`
: Overall between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.b.random`
: Overall between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`df.Q.b`
: Degrees of freedom for test of overall between subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.b.fixed`
: P-value of between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.b.random`
: P-value of between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`tau.w`
: Square-root of between-study variance within subgroups - if `byvar` is not missing.

`C.w` Scaling factor utilised internally to calculate common tau-squared across subgroups - if `byvar` is not missing.

`H.w` Heterogeneity statistic H within subgroups - if `byvar` is not missing.

`lower.H.w, upper.H.w`
Lower and upper confidence limti for heterogeneity statistic H within subgroups - if `byvar` is not missing.

`I2.w` Heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`lower.I2.w, upper.I2.w`
Lower and upper confidence limti for heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`incr.event` Increment added to number of events.

`keepdata` As defined above.

`data` Original data (set) used in function call (if `keepdata=TRUE`).

`subset` Information on subset of original data used in meta-analysis (if `keepdata=TRUE`).

`.glmm.fixed` GLMM object generated by call of `rma.glmm` function (fixed effect model).

`.glmm.random` GLMM object generated by call of `rma.glmm` function (random effects model).

`call` Function call.

`version` Version of R package **meta** used to create object.

`version.metafor`
Version of R package **metafor** used for GLMMs.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Agresti A & Coull BA (1998), Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*, **52**, 119–126.

DerSimonian R & Laird N (1986), Meta-analysis in clinical trials. *Controlled Clinical Trials*, **7**, 177–188.

Edward JM et al. (2006), Adherence to antiretroviral therapy in sub-saharan Africa and North America - a meta-analysis. *Journal of the American Medical Association*, **296**, 679–690.

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

Higgins JPT, Thompson SG, Spiegelhalter DJ (2009), A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society: Series A*, **172**, 137–159.

Knapp G & Hartung J (2003), Improved Tests for a Random Effects Meta-regression with a Single Covariate. *Statistics in Medicine*, **22**, 2693–2710, doi: 10.1002/sim.1482 .

Miller JJ (1978), The inverse of the Freeman-Tukey double arcsine transformation. *The American Statistician*, **32**, 138.

Newcombe RG (1998), Two-sided confidence intervals for the single proportion: Comparison of seven methods. *Statistics in Medicine*, **17**, 857–872.

Paule RC & Mandel J (1982), Consensus values and weighting factors. *Journal of Research of the National Bureau of Standards*, **87**, 377–385.

Pettigrew HM, Gart JJ, Thomas DG (1986), The bias and higher cumulants of the logarithm of a binomial variate. *Biometrika*, **73**, 425–435.

Stijnen T, Hamza TH, 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**, 3046–67.

Viechtbauer W (2010), Conducting Meta-Analyses in R with the Metafor Package. *Journal of Statistical Software*, **36**, 1–48.

See Also

update.meta, metacont, metagen, print.meta

Examples

```
# Apply various meta-analysis methods to estimate proportions
#
m1 <- metaprop(4:1, 10 * 1:4)
m2 <- update(m1, sm="PAS")
m3 <- update(m1, sm="PRAW")
m4 <- update(m1, sm="PLN")
m5 <- update(m1, sm="PFT")
#
m1
m2
m3
m4
m5
#
forest(m1)
# forest(m2)
# forest(m3)
# forest(m3, pscale=100)
# forest(m4)
# forest(m5)


# Do not back transform results, e.g. print logit transformed
# proportions if sm="PLOGIT" and store old settings
#
oldset <- settings.meta(backtransf=FALSE)
#
m6  <- metaprop(4:1, c(10, 20, 30, 40))
m7  <- update(m6, sm="PAS")
m8  <- update(m6, sm="PRAW")
m9  <- update(m6, sm="PLN")
m10 <- update(m6, sm="PFT")
#
forest(m6)
# forest(m7)
```

```
# forest(m8)
# forest(m8, pscale=100)
# forest(m9)
# forest(m10)
#
# Use old settings
#
settings.meta(oldset)


# Examples with zero events
#
m1 <- metaprop(c(0, 0, 10, 10), rep(100, 4))
m2 <- metaprop(c(0, 0, 10, 10), rep(100, 4), incr=0.1)
#
summary(m1)
summary(m2)
#
# forest(m1)
# forest(m2)


# Example from Miller (1978):
#
death <- c(3, 6, 10, 1)
animals <- c(11, 17, 21, 6)
#
m3 <- metaprop(death, animals, sm="PFT")
forest(m3)


# Data examples from Newcombe (1998)
# - apply various methods to estimate confidence intervals for
#   individual studies
#
event <- c(81, 15, 0, 1)
n <- c(263, 148, 20, 29)
#
m1 <- metaprop(event, n, sm="PLOGIT", method.ci="SA")
m2 <- update(m1, method.ci="SACC")
m3 <- update(m1, method.ci="WS")
m4 <- update(m1, method.ci="WSCC")
m5 <- update(m1, method.ci="CP")
#
lower <- round(rbind(NA, m1$lower, m2$lower, NA, m3$lower, m4$lower, NA, m5$lower), 4)
upper <- round(rbind(NA, m1$upper, m2$upper, NA, m3$upper, m4$upper, NA, m5$upper), 4)
#
tab1 <- data.frame(
  scen1=meta:::formatCI(lower[,1], upper[,1]),
  scen2=meta:::formatCI(lower[,2], upper[,2]),
  scen3=meta:::formatCI(lower[,3], upper[,3]),
  scen4=meta:::formatCI(lower[,4], upper[,4]),
  stringsAsFactors=FALSE
```

```
  )
names(tab1) <- c("r=81, n=263", "r=15, n=148", "r=0, n=20", "r=1, n=29")
row.names(tab1) <- c("Simple", "- SA", "- SACC",
                     "Score", "- WS", "- WSCC",
                     "Binomial", "- CP")
tab1[is.na(tab1)] <- ""
#
# Newcombe (1998), Table I, methods 1-5:
#
tab1
#
# Same confidence interval, i.e. unaffected by choice of summary measure
#
print(metaprop(event, n, sm="PLOGIT", method.ci="WS"), ma=FALSE)
print(metaprop(event, n, sm="PLN", method.ci="WS"), ma=FALSE)
print(metaprop(event, n, sm="PFT", method.ci="WS"), ma=FALSE)
print(metaprop(event, n, sm="PAS", method.ci="WS"), ma=FALSE)
print(metaprop(event, n, sm="PRAW", method.ci="WS"), ma=FALSE)
#
# Different confidence intervals as argument sm="NAsm"
#
print(metaprop(event, n, sm="PLOGIT", method.ci="NAsm"), ma=FALSE)
print(metaprop(event, n, sm="PLN", method.ci="NAsm"), ma=FALSE)
print(metaprop(event, n, sm="PFT", method.ci="NAsm"), ma=FALSE)
print(metaprop(event, n, sm="PAS", method.ci="NAsm"), ma=FALSE)
print(metaprop(event, n, sm="PRAW", method.ci="NAsm"), ma=FALSE)
#
# Different confidence intervals as argument backtransf=FALSE.
# Accordingly, method.ci="NAsm" used internally.
#
print(metaprop(event, n, sm="PLOGIT", method.ci="WS"), ma=FALSE, backtransf=FALSE)
print(metaprop(event, n, sm="PLN", method.ci="WS"), ma=FALSE, backtransf=FALSE)
print(metaprop(event, n, sm="PFT", method.ci="WS"), ma=FALSE, backtransf=FALSE)
print(metaprop(event, n, sm="PAS", method.ci="WS"), ma=FALSE, backtransf=FALSE)
print(metaprop(event, n, sm="PRAW", method.ci="WS"), ma=FALSE, backtransf=FALSE)
#
# Same results (printed on original and log scale, respectively)
#
print(metaprop(event, n, sm="PLN", method.ci="NAsm"), ma=FALSE)
print(metaprop(event, n, sm="PLN"), ma=FALSE, backtransf=FALSE)
# Results for first study (on log scale)
round(log(c(0.3079848, 0.2569522, 0.3691529)), 4)


# Meta-analysis using generalised linear mixed models
# (R packages 'metafor' and 'lme4' must be available)
#
# metaprop(event, n, method = "GLMM")


#
# Print results as events per 1000 observations
#
```

```
print(metaprop(6:8, c(100, 1200, 1000)), pscale = 1000, digits = 1)
```

`metarate` *Meta-analysis of single incidence rates*

Description

Calculation of an overall incidence rate from studies reporting a single incidence rate. Inverse variance method and generalised linear mixed model (GLMM) are available for pooling. For GLMMs, the `rma.glmm` function from R package **metafor** (Viechtbauer 2010) is called internally.

Usage

```
metarate(event, time, studlab,
         data=NULL, subset=NULL, exclude=NULL,
         method = "Inverse",
         sm=gs("smrate"),
         incr=gs("incr"), allincr=gs("allincr"),
         addincr=gs("addincr"),
         level=gs("level"), level.comb=gs("level.comb"),
         comb.fixed=gs("comb.fixed"), comb.random=gs("comb.random"),
         hakn=gs("hakn"),
         method.tau=
         ifelse(!is.na(charmatch(tolower(method), "glmm", nomatch = NA)),
                "ML", gs("method.tau")),
         tau.preset=NULL, TE.tau=NULL,
         tau.common=gs("tau.common"),
         prediction=gs("prediction"), level.predict=gs("level.predict"),
 null.effect=NA,
         method.bias=gs("method.bias"),
         backtransf=gs("backtransf"),
         irscale=1, irunit="person-years",
         title=gs("title"), complab=gs("complab"), outclab="",
         byvar, bylab, print.byvar=gs("print.byvar"),
         byseparator = gs("byseparator"),
         keepdata=gs("keepdata"),
         warn=gs("warn"),
         control=NULL,
 ...)
```

Arguments

`event`	Number of events.
`time`	Person time at risk.
`studlab`	An optional vector with study labels.
`data`	An optional data frame containing the study information, i.e., event and time.

subset	An optional vector specifying a subset of studies to be used.
exclude	An optional vector specifying studies to exclude from meta-analysis, however, to include in printouts and forest plots.
method	A character string indicating which method is to be used for pooling of studies. One of "Inverse" and "GLMM", can be abbreviated.
sm	A character string indicating which summary measure ("IR", "IRLN", "IRS", or "IRFT") is to be used for pooling of studies, see Details.
incr	A numeric which is added to the event number of studies with zero events, i.e., studies with an incidence rate of 0.
allincr	A logical indicating if incr is considered for all studies if at least one study has zero events. If FALSE (default), incr is considered only in studies with zero events.
addincr	A logical indicating if incr is used for all studies irrespective of number of events.
level	The level used to calculate confidence intervals for individual studies.
level.comb	The level used to calculate confidence intervals for pooled estimates.
comb.fixed	A logical indicating whether a fixed effect meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
prediction	A logical indicating whether a prediction interval should be printed.
level.predict	The level used to calculate prediction interval for a new study.
hakn	A logical indicating whether the method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
method.tau	A character string indicating which method is used to estimate the between-study variance τ^2, see Details.
tau.preset	Prespecified value for the square-root of the between-study variance τ^2.
TE.tau	Overall treatment effect used to estimate the between-study variance tau-squared.
tau.common	A logical indicating whether tau-squared should be the same across subgroups.
null.effect	A numeric value specifying the effect under the null hypothesis.
method.bias	A character string indicating which test is to be used. Either "rank", "linreg", or "mm", can be abbreviated. See function metabias.
backtransf	A logical indicating whether results for transformed rates (argument sm!="IR") should be back transformed in printouts and plots. If TRUE (default), results will be presented as incidence rates; otherwise transformed rates will be shown.
irscale	A numeric defining a scaling factor for printing of rates.
irunit	A character string specifying the time unit used to calculate rates, e.g. person-years.
title	Title of meta-analysis / systematic review.
complab	Comparison label.
outclab	Outcome label.

byvar	An optional vector containing grouping information (must be of same length as event).
bylab	A character string with a label for the grouping variable.
print.byvar	A logical indicating whether the name of the grouping variable should be printed in front of the group labels.
byseparator	A character string defining the separator between label and levels of grouping variable.
keepdata	A logical indicating whether original data (set) should be kept in meta object.
warn	A logical indicating whether the addition of incr to studies with zero events should result in a warning.
control	An optional list to control the iterative process to estimate the between-study variance tau^2. This argument is passed on to rma.uni or rma.glmm, respectively.
...	Additional arguments passed on to rma.glmm function.

Details

Fixed effect and random effects meta-analysis of single incidence rates to calculate an overall rate. The following transformations of incidence rates are implemented to calculate an overall rate:

- Log transformation (sm="IRLN", default)
- Square root transformation (sm="IRS")
- Freeman-Tukey Double arcsine transformation (sm="IRFT")
- No transformation (sm="IR")

Note, you should use R function metainc to compare incidence rates of pairwise comparisons instead of using metarate for each treatment arm separately which will break randomisation in randomised controlled trials.

Argument irscale can be used to rescale rates, e.g. irscale=1000 means that rates are expressed as events per 1000 time units, e.g. person-years. This is useful in situations with (very) low rates. Argument irunit can be used to specify the time unit used in individual studies (default: "person-years"). This information is printed in summaries and forest plots if argument irscale is not equal to 1.

For several arguments defaults settings are utilised (assignments using gs function). These defaults can be changed using the settings.meta function.

Internally, both fixed effect and random effects models are calculated regardless of values choosen for arguments comb.fixed and comb.random. Accordingly, the estimate for the random effects model can be extracted from component TE.random of an object of class "meta" even if argument comb.random=FALSE. However, all functions in R package **meta** will adequately consider the values for comb.fixed and comb.random. E.g. function print.meta will not print results for the random effects model if comb.random=FALSE.

A random intercept Poisson regression model can be utilised for the meta-analysis of incidence rates (Stijnen et al., 2010). This method is available (argument method = "GLMM") by calling the rma.glmm function from R package **metafor** internally.

If the summary measure is equal to "IR" or "IRLN", a continuity correction is applied if any study has zero events, i.e., an incidence rate of 0. By default, 0.5 is used as continuity correction (argument `incr`). This continuity correction is used both to calculate individual study results with confidence limits and to conduct meta-analysis based on the inverse variance method. For Freeman-Tukey and square root transformation and GLMMs no continuity correction is used.

Argument `byvar` can be used to conduct subgroup analysis for all methods but GLMMs. Instead use the `metareg` function for GLMMs which can also be used for continuous covariates.

A prediction interval for treatment effect of a new study is calculated (Higgins et al., 2009) if arguments `prediction` and `comb.random` are `TRUE`.

R function `update.meta` can be used to redo the meta-analysis of an existing metarate object by only specifying arguments which should be changed.

For the random effects, the method by Hartung and Knapp (2003) is used to adjust test statistics and confidence intervals if argument `hakn=TRUE`.

The DerSimonian-Laird estimate (1986) is used in the random effects model if `method.tau="DL"`. The iterative Paule-Mandel method (1982) to estimate the between-study variance is used if argument `method.tau="PM"`. Internally, R function `paulemandel` is called which is based on R function mpaule.default from R package **metRology** from S.L.R. Ellison <s.ellison at lgc.co.uk>.

If R package **metafor** (Viechtbauer 2010) is installed, the following methods to estimate the between-study variance τ^2 (argument `method.tau`) are also available:

- Restricted maximum-likelihood estimator (`method.tau="REML"`)
- Maximum-likelihood estimator (`method.tau="ML"`)
- Hunter-Schmidt estimator (`method.tau="HS"`)
- Sidik-Jonkman estimator (`method.tau="SJ"`)
- Hedges estimator (`method.tau="HE"`)
- Empirical Bayes estimator (`method.tau="EB"`).

For these methods the R function `rma.uni` of R package **metafor** is called internally. See help page of R function `rma.uni` for more details on these methods to estimate between-study variance.

Value

An object of class `c("metarate", "meta")` with corresponding `print`, `summary`, and `forest` functions. The object is a list containing the following components:

`event, n, studlab, exclude,`

`sm, incr, allincr, addincr, method.ci,`

`level, level.comb,`
: As defined above.

`comb.fixed, comb.random,`

`hakn, method.tau, tau.preset, TE.tau, null.effect,`

`method.bias, tau.common, title, complab, outclab,`

`byvar, bylab, print.byvar, byseparator, warn`

`TE, seTE` Estimated (un)transformed incidence rate and its standard error for individual studies.

`lower, upper` Lower and upper confidence interval limits for individual studies.

`zval, pval` z-value and p-value for test of treatment effect for individual studies.

`w.fixed, w.random` Weight of individual studies (in fixed and random effects model).

`TE.fixed, seTE.fixed` Estimated overall (un)transformed incidence rate and standard error (fixed effect model).

`lower.fixed, upper.fixed` Lower and upper confidence interval limits (fixed effect model).

`zval.fixed, pval.fixed` z-value and p-value for test of overall effect (fixed effect model).

`TE.random, seTE.random` Estimated overall (un)transformed incidence rate and standard error (random effects model).

`lower.random, upper.random` Lower and upper confidence interval limits (random effects model).

`zval.random, pval.random` z-value or t-value and corresponding p-value for test of overall effect (random effects model).

`prediction, level.predict` As defined above.

`seTE.predict` Standard error utilised for prediction interval.

`lower.predict, upper.predict` Lower and upper limits of prediction interval.

`k` Number of studies combined in meta-analysis.

`Q` Heterogeneity statistic Q.

`df.Q` Degrees of freedom for heterogeneity statistic.

`pval.Q` P-value of heterogeneity test.

`Q.LRT` Heterogeneity statistic for likelihood-ratio test (only if `method = "GLMM"`).

`df.Q.LRT` Degrees of freedom for likelihood-ratio test

`pval.Q.LRT` P-value of likelihood-ratio test.

`tau` Square-root of between-study variance.

`se.tau` Standard error of square-root of between-study variance.

`C` Scaling factor utilised internally to calculate common tau-squared across subgroups.

`method` A character string indicating method used for pooling: `"Inverse"`

`df.hakn` Degrees of freedom for test of treatment effect for Hartung-Knapp method (only if `hakn=TRUE`).

bylevs Levels of grouping variable - if byvar is not missing.

TE.fixed.w, seTE.fixed.w
Estimated treatment effect and standard error in subgroups (fixed effect model) - if byvar is not missing.

lower.fixed.w, upper.fixed.w
Lower and upper confidence interval limits in subgroups (fixed effect model) - if byvar is not missing.

zval.fixed.w, pval.fixed.w
z-value and p-value for test of treatment effect in subgroups (fixed effect model) - if byvar is not missing.

TE.random.w, seTE.random.w
Estimated treatment effect and standard error in subgroups (random effects model) - if byvar is not missing.

lower.random.w, upper.random.w
Lower and upper confidence interval limits in subgroups (random effects model) - if byvar is not missing.

zval.random.w, pval.random.w
z-value or t-value and corresponding p-value for test of treatment effect in subgroups (random effects model) - if byvar is not missing.

w.fixed.w, w.random.w
Weight of subgroups (in fixed and random effects model) - if byvar is not missing.

df.hakn.w Degrees of freedom for test of treatment effect for Hartung-Knapp method in subgroups - if byvar is not missing and hakn=TRUE.

n.harmonic.mean.w
Harmonic mean of number of observations in subgroups (for back transformation of Freeman-Tukey Double arcsine transformation) - if byvar is not missing.

event.w Number of events in subgroups - if byvar is not missing.

n.w Number of observations in subgroups - if byvar is not missing.

k.w Number of studies combined within subgroups - if byvar is not missing.

k.all.w Number of all studies in subgroups - if byvar is not missing.

Q.w.fixed Overall within subgroups heterogeneity statistic Q (based on fixed effect model) - if byvar is not missing.

Q.w.random Overall within subgroups heterogeneity statistic Q (based on random effects model) - if byvar is not missing (only calculated if argument tau.common is TRUE).

df.Q.w Degrees of freedom for test of overall within subgroups heterogeneity - if byvar is not missing.

pval.Q.w.fixed P-value of within subgroups heterogeneity statistic Q (based on fixed effect model) - if byvar is not missing.

pval.Q.w.random
P-value of within subgroups heterogeneity statistic Q (based on random effects model) - if byvar is not missing.

`Q.b.fixed` Overall between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`Q.b.random` Overall between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`df.Q.b` Degrees of freedom for test of overall between subgroups heterogeneity - if `byvar` is not missing.

`pval.Q.b.fixed` P-value of between subgroups heterogeneity statistic Q (based on fixed effect model) - if `byvar` is not missing.

`pval.Q.b.random`
P-value of between subgroups heterogeneity statistic Q (based on random effects model) - if `byvar` is not missing.

`tau.w` Square-root of between-study variance within subgroups - if `byvar` is not missing.

`C.w` Scaling factor utilised internally to calculate common tau-squared across subgroups - if `byvar` is not missing.

`H.w` Heterogeneity statistic H within subgroups - if `byvar` is not missing.

`lower.H.w, upper.H.w`
Lower and upper confidence limti for heterogeneity statistic H within subgroups - if `byvar` is not missing.

`I2.w` Heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`lower.I2.w, upper.I2.w`
Lower and upper confidence limti for heterogeneity statistic I2 within subgroups - if `byvar` is not missing.

`incr.event` Increment added to number of events.

`keepdata` As defined above.

`data` Original data (set) used in function call (if `keepdata=TRUE`).

`subset` Information on subset of original data used in meta-analysis (if `keepdata=TRUE`).

`.glmm.fixed` GLMM object generated by call of `rma.glmm` function (fixed effect model).

`.glmm.random` GLMM object generated by call of `rma.glmm` function (random effects model).

`call` Function call.

`version` Version of R package **meta** used to create object.

`version.metafor`
Version of R package **metafor** used for GLMMs.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

DerSimonian R & Laird N (1986), Meta-analysis in clinical trials. *Controlled Clinical Trials*, **7**, 177–188.

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

Higgins JPT, Thompson SG, Spiegelhalter DJ (2009), A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society: Series A*, **172**, 137–159.

Knapp G & Hartung J (2003), Improved Tests for a Random Effects Meta-regression with a Single Covariate. *Statistics in Medicine*, **22**, 2693–2710, doi: 10.1002/sim.1482 .

Paule RC & Mandel J (1982), Consensus values and weighting factors. *Journal of Research of the National Bureau of Standards*, **87**, 377–385.

Stijnen T, Hamza TH, 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**, 3046–67.

Viechtbauer W (2010), Conducting Meta-Analyses in R with the Metafor Package. *Journal of Statistical Software*, **36**, 1–48.

See Also

`update.meta`, `metacont`, `metagen`, `print.meta`

Examples

```
#
# Apply various meta-analysis methods to estimate incidence rates
#
m1 <- metarate(4:1, c(10, 20, 30, 40))
m2 <- update(m1, sm="IR")
m3 <- update(m1, sm="IRS")
m4 <- update(m1, sm="IRFT")
#
m1
m2
m3
m4
#
forest(m1)
forest(m1, irscale=100)
forest(m1, irscale=100, irunit="person-days")
forest(m1, backtransf = FALSE)
# forest(m2)
# forest(m3)
# forest(m4)
#
m5 <- metarate(40:37, c(100, 200, 300, 400), sm="IRFT")
m5
```

`metareg` *Meta-regression*

Description

Meta-regression for objects of class `meta`. This is a wrapper function for the R function `rma.uni` in the R package **metafor** (Viechtbauer 2010).

Usage

```
metareg(x, formula, method.tau = x$method.tau, hakn = x$hakn,
        level.comb = x$level.comb, intercept = TRUE,...)
```

Arguments

x	An object of class meta.
formula	Either a character string or a formula object.
method.tau	A character string indicating which method is used to estimate the between-study variance tau-squared. Either "FE", "DL", "REML", "ML", "HS", "SJ", "HE", or "EB", can be abbreviated.
hakn	A logical indicating whether the method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
level.comb	The level used to calculate confidence intervals for parameter estimates in the meta-regression model.
intercept	A logical indicating whether an intercept should be included in the meta-regression model.
...	Additional arguments passed to R function rma.uni.

Details

This R function is a wrapper function for R function rma.uni in the R package **metafor** (Viechtbauer 2010), i.e., function metareg can only be used if R package **metafor** is installed.

Note, results are not back-transformed in printouts of meta-analyses using summary measures with transformations, e.g., log risk ratios are printed instead of the risk ratio if argument sm="RR" and logit transformed proportions are printed if argument sm="PLOGIT".

Argument '...' can be used to pass additional arguments to R function rma.uni. For example, argument control to provide a list of control values for the iterative estimation algorithm. See help page of R function rma.uni for more details.

Value

An object of class c("metareg", "rma.uni","rma"). Please look at the help page of R function rma.uni for more details on the output from this function.

In addition, a list .meta is added to the output containing the following components:

x, formula, method.tau, hakn, level.comb, intercept	As definied above.
dots	Information provided in argument '...'.
call	Function call.
version	Version of R package **meta** used to create object.
version.metafor	Version of R package **metafor** used to create object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Viechtbauer W (2010), Conducting Meta-Analyses in R with the Metafor Package. *Journal of Statistical Software*, **36**, 1–48.

See Also

bubble, summary.meta, metagen

Examples

```
data(Fleiss93cont)

# Add some (fictitious) grouping variables:
Fleiss93cont$age <- c(55, 65, 55, 65, 55)
Fleiss93cont$region <- c("Europe", "Europe", "Asia", "Asia", "Europe")

meta1 <- metacont(n.e, mean.e, sd.e,
                  n.c, mean.c, sd.c,
                  data = Fleiss93cont, sm = "MD")


mu1 <- update(meta1, byvar = region)

mu2 <- update(meta1, byvar = region,
              tau.common = TRUE, comb.fixed = FALSE)

## Not run:
# Warnings due to wrong ordering of arguments (order has changed with
# version 3.0-0 of R package meta)
#
metareg(~ region, meta1)
metareg(~ region, data = meta1)

# Warning as no information on covariate is available
#
metareg(meta1)

## End(Not run)

# Do meta-regression for covariate region
# (see R code to create object mu2)
#
metareg(mu2)

# Same result for
# - tau-squared
# - test of heterogeneity
# - test for subgroup differences
```

```
# (as argument 'tau.common' was used to create mu2)
#
mu2
metareg(mu2, intercept = FALSE)
metareg(meta1, region)
#
# Different result for
# - tau-squared
# - test of heterogeneity
# - test for subgroup differences
# (as argument 'tau.common' is - by default - FALSE)
#
mu1

# Generate bubble plot
#
bubble(metareg(mu2))

# Do meta-regression with two covariates
#
metareg(mu1, region + age)

# Do same meta-regressions using 'official' formula notation
#
metareg(meta1, ~ region)
metareg(mu1, ~ region + age)

# Do meta-regression using REML method and print intermediate results
# for iterative estimation algorithm; furthermore print results with
# three digits.
#
metareg(mu1, region, method.tau = "REML",
        control = list(verbose = TRUE), digits = 3)

# Use Hartung-Knapp method
#
mu3 <- update(mu2, hakn = TRUE)
mu3
metareg(mu3, intercept = FALSE)
```

Olkin95 *Thrombolytic Therapy after Acute Myocardial Infarction*

Description

Meta-analysis on Thrombolytic Therapy after Acute Myocardial Infarction

Usage

```
data(Olkin95)
```

Format

A data frame with the following columns:

author First author

year Year of publication

event.e Number of events in experimental group

n.e Number of observations in experimental group

event.c Number of events in control group

n.c Number of observations in control group

Source

Olkin I (1995), Statistical and theoretical considerations in meta-analysis. *Journal of Clinical Epidemiology*, **48**, 133–146.

Examples

```
data(Olkin95)
summary(metabin(event.e, n.e, event.c, n.c, data=Olkin95))
```

`print.meta` *Print and summary method for objects of class meta*

Description

Print and summary method for objects of class `meta`.

Usage

```
## S3 method for class 'meta'
print(x, sortvar,
        comb.fixed=x$comb.fixed,
        comb.random=x$comb.random,
        prediction=x$prediction,
        details=FALSE, ma=TRUE, backtransf=x$backtransf,
        pscale=x$pscale, irscale=x$irscale, irunit=x$irunit,
        digits=gs("digits"), digits.se=gs("digits.se"),
        digits.tau2=gs("digits.tau2"), digits.I2=gs("digits.I2"),
        digits.prop=gs("digits.prop"), digits.weight=gs("digits.weight"),
big.mark=gs("big.mark"),
        warn.backtransf=FALSE,
        ...)

## S3 method for class 'metabias'
print(x, ...)
```

```
## S3 method for class 'meta'
summary(object,
        comb.fixed=object$comb.fixed, comb.random=object$comb.random,
        prediction=object$prediction,
        backtransf=object$backtransf,
        pscale=object$pscale, irscale=object$irscale, irunit=object$irunit,
        bylab=object$bylab, print.byvar=object$print.byvar,
        byseparator=object$byseparator, bystud=FALSE,
        print.CMH=object$print.CMH, warn=object$warn, ...)

## S3 method for class 'summary.meta'
print(x,
        comb.fixed=x$comb.fixed, comb.random=x$comb.random,
        prediction=x$prediction,
        print.byvar=x$print.byvar, byseparator=x$byseparator,
        print.CMH=x$print.CMH,
        header=TRUE, backtransf=x$backtransf,
        pscale=x$pscale, irscale=x$irscale, irunit=x$irunit,
        bylab.nchar=35,
        digits=gs("digits"),
        digits.zval=gs("digits.zval"),
        digits.pval=max(gs("digits.pval"), 2),
        digits.pval.Q=max(gs("digits.pval.Q"), 2),
        digits.Q=gs("digits.Q"), digits.tau2=gs("digits.tau2"),
        digits.H=gs("digits.H"), digits.I2=gs("digits.I2"),
scientific.pval=gs("scientific.pval"), big.mark=gs("big.mark"),
        print.I2=gs("print.I2"), print.H=gs("print.H"),
        print.Rb=gs("print.Rb"),
        text.tau2=gs("text.tau2"), text.I2=gs("text.I2"),
        text.Rb=gs("text.Rb"),
        warn.backtransf=FALSE,
        ...)

cilayout(bracket="[", separator="; ")
```

Arguments

x	An object of class meta, metabias, or summary.meta.
object	An object of class meta.
sortvar	An optional vector used to sort the individual studies (must be of same length as x$TE).
comb.fixed	A logical indicating whether a fixed effect meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
prediction	A logical indicating whether a prediction interval should be printed.
bylab	A character string with a label for the grouping variable.

print.byvar	A logical indicating whether the name of the grouping variable should be printed in front of the group labels.
byseparator	A character string defining the separator between label and levels of grouping variable.
header	A logical indicating whether information on title of meta-analysis, comparison and outcome should be printed at the beginning of the printout.
details	A logical indicating whether further details of individual studies should be printed.
ma	A logical indicating whether the summary results of the meta-analysis should be printed.
backtransf	A logical indicating whether printed results should be back transformed. If backtransf=TRUE, results for sm="OR" are printed as odds ratios rather than log odds ratios and results for sm="ZCOR" are printed as correlations rather than Fisher's z transformed correlations, for example.
pscale	A numeric giving scaling factor for printing of single event probabilities or risk differences, i.e. if argument sm is equal to "PLOGIT", "PLN", "PRAW", "PAS", "PFT", or "RD".
irscale	A numeric defining a scaling factor for printing of single incidence rates or incidence rate differences, i.e. if argument sm is equal to "IR", "IRLN", "IRS", "IRFT", or "IRD".
irunit	A character specifying the time unit used to calculate rates, e.g. person-years.
bylab.nchar	A numeric specifying the number of characters to print from label for the grouping variable.
bystud	A logical indicating whether results of individual studies should be printed by grouping variable.
print.CMH	A logical indicating whether result of the Cochran-Mantel-Haenszel test for overall effect should be printed.
digits	Minimal number of significant digits, see print.default.
digits.se	Minimal number of significant digits for standard deviations and standard errors, see print.default.
digits.zval	Minimal number of significant digits for z- or t-value, see print.default.
digits.pval	Minimal number of significant digits for p-value of overall treatment effect, see print.default.
digits.pval.Q	Minimal number of significant digits for p-value of heterogeneity test, see print.default.
digits.Q	Minimal number of significant digits for heterogeneity statistic Q, see print.default.
digits.tau2	Minimal number of significant digits for between-study variance, see print.default.
digits.H	Minimal number of significant digits for H statistic, see print.default.
digits.I2	Minimal number of significant digits for I-squared and Rb statistic, see print.default.
digits.prop	Minimal number of significant digits for proportions, see print.default.
digits.weight	Minimal number of significant digits for weights, see print.default.
scientific.pval	A logical specifying whether p-values should be printed in scientific notation, e.g., 1.2345e-01 instead of 0.12345.

big.mark	A character used as thousands separator.
print.I2	A logical specifying whether heterogeneity statistic I^2 should be printed.
warn	A logical indicating whether the use of summary.meta in connection with metacum or metainf should result in a warning.
warn.backtransf	A logical indicating whether a warning should be printed if backtransformed proportions and rates are below 0 and backtransformed proportions are above 1.
bracket	A character with bracket symbol to print lower confidence interval: "[", "(", "{", "".
separator	A character string with information on separator between lower and upper confidence interval.
print.H	A logical specifying whether heterogeneity statistic H should be printed.
print.Rb	A logical specifying whether heterogeneity statistic Rb should be printed.
text.tau2	Text printed to identify between-study variance tau^2.
text.I2	Text printed to identify heterogeneity statistic I^2.
text.Rb	Text printed to identify heterogeneity statistic Rb.
...	In print.meta, additional arguments are passed on to print.summary.meta called internally; otherwise, this argument is ignored.

Details

Note, in R package **meta**, version 3.0-0 some arguments have been removed from R functions summary.meta (arguments: byvar, level, level.comb, level.prediction) and print.summary.meta (arguments: level, level.comb, level.prediction). This functionality is now provided by R function update.meta (or directly in meta-analysis functions, e.g., metabin, metacont, metagen, metacor, and metaprop).

Review Manager 5 (RevMan 5) is the current software used for preparing and maintaining Cochrane Reviews (http://community.cochrane.org/tools/review-production-tools/revman-5). In RevMan 5, subgroup analyses can be defined and data from a Cochrane review can be imported to R using the function read.rm5. If a meta-analysis is then conducted using function metacr, information on subgroups is available in R (components byvar, bylab, and print.byvar, byvar in an object of class "meta"). Accordingly, by using function metacr there is no need to define subgroups in order to redo the statistical analysis conducted in the Cochrane review.

Note, for an object of type metaprop, starting with version 3.7-0 of meta, list elements TE, lower and upper in element study correspond to transformed proportions and confidence limits (regardless whether exact confidence limits are calculated; argument ciexact=TRUE in metaprop function). Accordingly, the following results are based on the same transformation defined by argument sm: list elements TE, lower and upper in elements study, fixed, random, within.fixed and within.random.

R function cilayout can be utilised to change the layout to print confidence intervals (both in printout from print.meta and print.summary.meta function as well as in forest plots). The default layout is "[lower; upper]". Another popular layout is "(lower - upper)" which is used throughout an R session by using R command cilayout("(", " - ").

Argument pscale can be used to rescale single proportions or risk differences, e.g. pscale=1000 means that proportions are expressed as events per 1000 observations. This is useful in situations with (very) low event probabilities.

Argument irscale can be used to rescale single rates or rate differences, e.g. irscale=1000 means that rates are expressed as events per 1000 time units, e.g. person-years. This is useful in situations with (very) low rates. Argument irunit can be used to specify the time unit used in individual studies (default: "person-years"). This information is printed in summaries and forest plots if argument irscale is not equal to 1.

Value

A list is returned by the function summary.meta with the following elements:

study	Results for individual studies (a list with elements TE, seTE, lower, upper, z, p, level, df).
fixed	Results for fixed effect model (a list with elements TE, seTE, lower, upper, z, p, level, df).
random	Results for random effects model (a list with elements TE, seTE, lower, upper, z, p, level, df).
k	Number of studies combined in meta-analysis.
Q	Heterogeneity statistic Q.
tau	Square-root of between-study variance.
se.tau	Standard error of square-root of between-study variance.
C	Scaling factor utilised internally to calculate common tau-squared across subgroups.
H	Heterogeneity statistic H (a list with elements TE, lower, upper).
I2	Heterogeneity statistic I2 (a list with elements TE, lower, upper), see Higgins & Thompson (2002).
Rb	Heterogeneity statistic Rb (a list with elements TE, lower, upper), see Crippa et al. (2016).
k.all	Total number of studies.
Q.CMH	Cochran-Mantel-Haenszel test statistic for overall effect.
sm	A character string indicating underlying summary measure.
method	A character string with the pooling method.
call	Function call.
ci.lab	Label for confidence interval.
hakn	A logical indicating whether method by Hartung and Knapp was used.
method.tau	A character string indicating which method is used to estimate the between-study variance tau-squared.
tau.common	A logical indicating whether tau-squared is assumed to be the same across subgroups.
within.fixed	Result for fixed effect model within groups (a list with elements TE, seTE, lower, upper, z, p, level, df, harmonic.mean) - if byvar is not missing.

`within.random` Result for random effects model within groups (a list with elements TE, seTE, lower, upper, z, p, level, df, harmonic.mean) - if `byvar` is not missing.

`k.w` Number of studies combined within groups - if `byvar` is not missing.

`Q.w` Heterogeneity statistic Q within groups - if `byvar` is not missing.

`Q.b.fixed` Heterogeneity statistic Q between groups (based on fixed effect model) - if `byvar` is not missing.

`Q.b.random` Heterogeneity statistic Q between groups (based on random effects model) - if `byvar` is not missing.

`tau.w` Square-root of between-study variance within subgroups - if `byvar` is not missing.

`C.w` Scaling factor utilised internally to calculate common tau-squared across subgroups.

`H.w` Heterogeneity statistic H within subgroups (a list with elements TE, lower, upper) - if `byvar` is not missing.

`I2.w` Heterogeneity statistic I2 within subgroups (a list with elements TE, lower, upper) - if `byvar` is not missing.

`Rb.w` Heterogeneity statistic Rb within subgroups (a list with elements TE, lower, upper) - if `byvar` is not missing.

`H.resid` Statistic H for residual heterogeneity (a list with elements TE, lower, upper) - if `byvar` is not missing.

`I2.resid` Statistic I2 for residual heterogeneity (a list with elements TE, lower, upper) - if `byvar` is not missing.

`bylevs` Levels of grouping variable - if `byvar` is not missing.

`title` Title of meta-analysis / systematic review.

`complab` Comparison label.

`outclab` Outcome label.

`data` Original data (set) used to create meta object.

`subset` Information on subset of original data used in meta-analysis.

`prediction, level.predict`

`comb.fixed, comb.random, print.CMH`
As defined above.

`version` Version of R package **meta** used to create object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Cooper H & Hedges LV (1994), *The Handbook of Research Synthesis*. Newbury Park, CA: Russell Sage Foundation.

Crippa A, Khudyakov P, Wang M, Orsini N, Spiegelman D (2016), A new measure of between-studies heterogeneity in meta-analysis. *Statistics in Medicine*, **35**, 3661–75.

Higgins JPT & Thompson SG (2002), Quantifying heterogeneity in a meta-analysis. *Statistics in Medicine*, **21**, 1539–58.

See Also

update.meta, metabin, metacont, metagen

Examples

```
data(Fleiss93cont)
meta1 <- metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c,
                  data=Fleiss93cont, sm="SMD",
                  studlab=paste(study, year))
summary(meta1)

summary(update(meta1, byvar=c(1,2,1,1,2), bylab="group"))

forest(update(meta1, byvar=c(1,2,1,1,2), bylab="group"))

## Not run:
# Use unicode characters to print tau^2 and I^2
#
print(summary(meta1), text.tau2="\u03c4\u00b2", text.I2="I\u00b2")

## End(Not run)
```

---

print.rm5 *Print and summary methods for objects of class rm5*

Description

Print and summary methods for objects of class rm5.

Usage

```
## S3 method for class 'rm5'
print(x, ...)

## S3 method for class 'summary.rm5'
print(x, ...)

## S3 method for class 'rm5'
```

```
summary(object, comp.no, outcome.no, ...)

## S3 method for class 'rm5'
metabias(x, comp.no, outcome.no,
         method.bias="linreg",
         method.bias.binary=method.bias,
         method.bias.or="score",
         k.min=10, ...)
```

Arguments

x	An object of class rm5.
object	An object of class rm5.
comp.no	Comparison number.
outcome.no	Outcome number.
method.bias	A character string indicating which test for small-study effects is to be used for all outcomes. Either "rank", "linreg", or "mm", can be abbreviated. See function metabias
method.bias.binary	A character string indicating which test is to be used for binary outcomes. Either "rank", "linreg", "mm", "count", "score", or "peters", can be abbreviated. See function metabias
method.bias.or	A character string indicating which test is to be used for binary outcomes with odds ratio as summary measure. Either "rank", "linreg", "mm", "count", "score", or "peters", can be abbreviated. See function metabias
k.min	Minimum number of studies to perform test for small-study effects.
...	Additional arguments (ignored at the moment)

Details

Review Manager 5 (RevMan 5) is the current software used for preparing and maintaining Cochrane Reviews (http://community.cochrane.org/tools/review-production-tools/revman-5). In RevMan 5, subgroup analyses can be defined and data from a Cochrane review can be imported to R using the function read.rm5.

The R function summary.rm5 can be used to redo all meta-analyses of the imported Cochrane Review.

The R function metabias.rm5 can be used to conduct a test for funnel plot asymmetry for all meta-analyses of the imported Cochrane Review.

The R function metacr is called internally.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Higgins, J.P.T and S. Green (2011), *Cochrane Handbook for Systematic Reviews of Interventions Version 5.1.0 [Updated March 2011]*. The Cochrane Library: http://www.cochrane-handbook.org

See Also

metabias.meta, summary.meta, read.rm5

Examples

```
# Locate export data file "Fleiss93_CR.csv"
# in sub-directory of package "meta"
#
filename <- system.file("data/Fleiss93_CR.csv.gz", package = "meta")
#
Fleiss93_CR <- read.rm5(filename)

#
# Print summary results for all meta-analysis:
#
summary(Fleiss93_CR)

#
# Print results for tests of small-study effects:
#
metabias(Fleiss93_CR, k.min=5)
```

---

read.mtv *Import RevMan 4 data files (.mtv)*

Description

Reads a file created with RevMan 4 and creates a data frame from it.

Usage

```
read.mtv(file)
```

Arguments

file	The name of a file to read data values from.

Details

Reads a file created with RevMan 4 (Menu: "File" - "Export" - "Analysis data file...") and creates a data frame from it.

Value

A data frame containing the following components:

comp.no	Comparison number.
outcome.no	Outcome number.
group.no	Group number.
studlab	Study label.
year	Year of publication.
event.e	Number of events in experimental group.
n.e	Number of observations in experimental group.
event.c	Number of events in control group.
n.c	Number of observations in control group.
mean.e	Estimated mean in experimental group.
sd.e	Standard deviation in experimental group.
mean.c	Estimated mean in control group.
sd.c	Standard deviation in control group.
O.E	Observed minus expected (IPD analysis).
V	Variance of O.E (IPD analysis).
order	Ordering of studies.
conceal	Concealment of treatment allocation.
grplab	Group label.
type	Type of outcome. D = dichotomous, C = continuous, P = IPD.
outclab	Outcome label.
graph.exp	Graph label for experimental group.
graph.cont	Graph label for control group.
label.exp	Label for experimental group.
label.cont	Label for control group.
complab	Comparison label.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Review Manager (RevMan) [Computer program]. Version 4.2 for Windows. Copenhagen: The Nordic Cochrane Centre, The Cochrane Collaboration, 2003.

See Also

metabin, metacont, metagen

Examples

```
# Locate MTV-data file "FLEISS93.MTV" in sub-directory of package "meta"
#
filename <- system.file("extdata/Fleiss93.MTV", package = "meta")
#
fleiss93.cc <- read.mtv(filename)

# Same result as R Command example(Fleiss93):
#
metabin(event.e, n.e, event.c, n.c,
        data=fleiss93.cc, subset=type=="D",
        studlab=paste(studlab, year))

# Same result: example(Fleiss93cont)
#
metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c,
         data=fleiss93.cc, subset=type=="C",
         studlab=paste(studlab, year))
```

`read.rm5` *Import RevMan 5 data files (.csv)*

Description

Reads data file from Cochrane Intervention review created with RevMan 5 and creates a data frame from it.

Usage

```
read.rm5(file, sep=",", quote = "\"", title,
         numbers.in.labels=TRUE)
```

Arguments

`file`	The name of a file to read data values from.
`sep`	The field separator character. Values on each line of the file are separated by this character. The comma is the default field separator character in RevMan 5.
`quote`	The set of quoting characters. In RevMan 5 a "\"" is the default quoting character.
`title`	Title of Cochrane review.
`numbers.in.labels`	A logical indicating whether comparision number and outcome number should be printed at the beginning of the comparison (argument `complab`) and outcome label (argument `outclab`); this is the default in RevMan 5.

Details

Review Manager 5 (RevMan 5) is the current software used for preparing and maintaining Cochrane Reviews (http://community.cochrane.org/tools/review-production-tools/revman-5). RevMan 5 includes the ability to write Systematic reviews of interventions, Diagnostic test accuracy reviews, Methodology reviews and Overviews of reviews.

This function provides the ability to read a data file from a Cochrane Intervention review created with RevMan 5; a data frame is created from it. Cochrane Intervention reviews are based on the comparison of two interventions.

In order to generate a data analysis file in RevMan 5 use the following Menu points: "File" - "Export" - "Data and analyses". It is mandatory to include the following fields in the exported data file by selecting them with the mouse cursor in the Export Analysis Data Wizard: (i) Comparison Number, (ii) Outcome Number, (iii) Subgroup Number. When these fields are not selected a corresponding error message will be printed in R. It is recommended to include all fields in the exported data file except for the last field "Risk of bias tables". For example, in order to redo the meta-analysis in R for the RevMan 5 data type "O-E and Variance" the fields "O-E" and "Variance" have to be selected in the Export Analysis Data Wizard. If the last field "Risk of bias tables" is selected the import in R fails with an error message "line X did not have Y elements".

By default in RevMan 5, the name of the exported data file is the title of the Cochrane Review. Accordingly, information on the title is extracted from the name of the exported data file (argument: `file`) if argument `title` is missing (default).

Each respective meta-analysis for arguments `event.e.pooled` – `df.pooled` is defined by values for `"comp.no"` and `"outcome.no"`, and `"grp.no"`.

Value

A data frame containing the following components:

`comp.no`	Comparison number.
`outcome.no`	Outcome number.
`group.no`	Group number.
`studlab`	Study label.
`year`	Year of publication.
`event.e`	Number of events in experimental group.
`n.e`	Number of observations in experimental group.
`event.c`	Number of events in control group.
`n.c`	Number of observations in control group.
`mean.e`	Estimated mean in experimental group.
`sd.e`	Standard deviation in experimental group.
`mean.c`	Estimated mean in control group.
`sd.c`	Standard deviation in control group.
`O.E`	Observed minus expected (IPD analysis).
`V`	Variance of `O.E` (IPD analysis).
`TE, seTE`	Estimated treatment effect and standard error of individual studies.

lower, upper	Lower and upper limit of 95% confidence interval for treatment effect in individual studies.
weight	Weight of individual studies (according to meta-analytical method used in respective meta-analysis - see below for details).
order	Ordering of studies.
grplab	Group label.
type	Type of outcome. D = dichotomous, C = continuous, P = IPD.
method	A character string indicating which method has been used for pooling of studies. One of "Inverse", "MH", or "Peto".
sm	A character string indicating which summary measure has been used for pooling of studies.
model	A character string indicating which meta-analytical model has been used (either "Fixed" or "Random").
comb.fixed	A logical indicating whether fixed effect meta-analysis has been used in respective meta-analysis (see below for details).
comb.random	A logical indicating whether random effects meta-analysis has been used in respective meta-analysis (see below for details).
outclab	Outcome label.
k	Total number of studies combined in respective meta-analysis).
event.e.pooled	Number of events in experimental group in respective meta-analysis (see below for details).
n.e.pooled	Number of observations in experimental group in respective meta-analysis (see below for details).
event.c.pooled	Number of events in control group in respective meta-analysis (see below for details).
n.c.pooled	Number of observations in control group in respective meta-analysis (see below for details).
TE.pooled	Estimated treatment effect in respective meta-analysis (see below for details).
lower, upper	Lower and upper limit of 95% confidence interval for treatment effect in respective meta-analysis (see below for details).
weight.pooled	Total weight in respective meta-analysis (see below for details).
Z.pooled	Z-score for test of overall treatment effect in respective meta-analysis (see below for details).
pval.pooled	P-value for test of overall treatment effect in respective meta-analysis (see below for details).
Q	Heterogeneity statistic Q in respective meta-analysis (see below for details).
pval.Q	P-value of heterogeneity statistic Q in respective meta-analysis (see below for details).
I2	Heterogeneity statistic I2 in respective meta-analysis (see below for details).
tau2	Between-study variance (moment estimator of DerSimonian-Laird) in respective meta-analysis.

Q.w	Heterogeneity statistic Q within groups in respective meta-analysis (see below for details).
pval.Q.w	P-value of heterogeneity statistic Q within groups in respective meta-analysis (see below for details).
I2.w	Heterogeneity statistic I2 within groups in respective meta-analysis (see below for details).
label.e	Label for experimental group.
label.c	Label for control group.
label.left	Graph label on left side of forest plot.
label.right	Graph label on right side of forest plot.
RR.cochrane	A logical indicating if 2*incr instead of 1*incr is to be added to n.e and n.c in the calculation of the risk ratio (i.e., sm="RR") for studies with a zero cell. This is used in RevMan 5.
complab	Comparison label.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Review Manager (RevMan) [Computer program]. Version 5.3. Copenhagen: The Nordic Cochrane Centre, The Cochrane Collaboration, 2014.

See Also

metabin, metacont, metagen, metacr

Examples

```
# Locate export data file "Fleiss93_CR.csv"
# in sub-directory of package "meta"
#
filename <- system.file("data/Fleiss93_CR.csv.gz", package = "meta")
#
Fleiss93_CR <- read.rm5(filename)

# Same result as R command example(Fleiss93):
#
metacr(Fleiss93_CR)

# Same result as R command example(Fleiss93cont):
#
metacr(Fleiss93_CR, 1, 2)
```

`settings.meta` *Print and change default settings to conduct and print or plot meta-analyses in R package* ***meta****.*

Description

Print and change default settings to conduct and print or plot meta-analyses in R package **meta**. The following general settings are available: *Review Manager 5*, *Journal of the American Medical Association*.

Usage

```
settings.meta(...)
```

Arguments

`...`	Arguments to change default settings.

Details

This function can be used to define defaults for several arguments (i.e., assignments using `gs`) of the following R functions: `metabin`, `metacont`, `metacor`, `metacr`, `metagen`, `metainc`, `metaprop`, `metarate`

Furthermore, some of these settings are considered to print meta-analysis results using `print.meta` and `print.summary.meta`, and to produce forest plots using `forest.meta`.

The function can be used to either change individual settings (see Examples) or use one of the following general settings:

- `settings.meta("revman5")`
- `settings.meta("jama")`

The first command can be used to reproduce meta-analyses from Cochrane reviews conducted with *Review Manager 5* (RevMan 5, http://community.cochrane.org/tools/review-production-tools/revman-5) and specifies to use a RevMan 5 layout in forest plots. The second command can be used to generate forest plots following instructions for authors of the *Journal of the American Medical Association* (http://jamanetwork.com/journals/jama/pages/instructions-for-authors).

RevMan 5 settings, in detail:

Argument	Value	Comment
`hakn`	FALSE	method not available in RevMan 5
`method.tau`	"DL"	only available method in RevMan 5
`tau.common`	FALSE	common between-study variance in subgroups
`MH.exact`	FALSE	exact Mantel-Haenszel method
`RR.cochrane`	TRUE	calculation of risk ratios
`layout`	"RevMan5"	layout for forest plots
`test.overall`	TRUE	print information on test of overall effect

digits.I2	0	number of digits for I-squared measure
digits.tau2	0	number of digits for tau-squared
CIbracket,	"["	
CIseparator	", "	print confidence intervals as "[., .]"

JAMA settings:

Argument	Value	Comment
layout	"JAMA"	layout for forest plots
test.overall	TRUE	print information on test of overall effect
digits.I2	0	number of digits for I-squared measure
CIbracket,	"("	
CIseparator	"-"	print confidence intervals as "(.-.)"

A list of all arguments with current settings is printed using the command `settings.meta("print")`.

In order to reset all settings of R package **meta** the command `settings.meta("reset")` can be used.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

gs, forest.meta

Examples

```
#
# Get listing of current settings
#
settings.meta("print")

#
# Meta-analyses using default settings
#
metabin(10, 20, 15, 20)
metaprop(4, 20)
metabin(10, 20, 15, 20, sm="RD")
metaprop(4, 20, sm="PLN")
#
# Change summary measure for R functions metabin and metaprop
# and store old settings
#
oldset <- settings.meta(smbin="RD", smprop="PLN")
#
metabin(10, 20, 15, 20)
metaprop(4, 20)
#
# Use old settings
```

```
#
settings.meta(oldset)

#
# Change level used to calculate confidence intervals
# (99%-CI for studies, 99.9%-CI for pooled effects)
#
metagen(1:3, (2:4)/10, sm="MD")
settings.meta(level=0.99, level.comb=0.999)
metagen(1:3, (2:4)/10, sm="MD")

#
# Always print a prediction interval
#
settings.meta(prediction=TRUE)
metagen(1:3, (2:4)/10, sm="MD")
metagen(4:6, (4:2)/10, sm="MD")

#
# Try to set unknown argument results in a warning
#
settings.meta(unknownarg=TRUE)

#
# Reset to default settings of R package meta
#
settings.meta("reset")
metabin(10, 20, 15, 20)
metaprop(4, 20)
metagen(1:3, (2:4)/10, sm="MD")

#
# Do not back transform results (e.g. print log odds ratios instead of
# odds ratios, print transformed correlations/proportions instead of
# correlations/proportions)
#
settings.meta(backtransf=FALSE)
metabin(10, 20, 15, 20)
metaprop(4, 20)
metacor(c(0.85, 0.7, 0.95), c(20, 40, 10))

#
# Forest plot using RevMan 5 style
#
settings.meta("revman5")
forest(metagen(1:3, (2:4)/10, sm="MD", comb.fixed=FALSE),
       label.left="Favours A", label.right="Favours B",
       colgap.studlab = grid::unit(2, "cm"),
       colgap.forest.left = grid::unit(0.2, "cm"))

#
# Forest plot using JAMA style
#
```

```
settings.meta("jama")
forest(metagen(1:3, (2:4)/10, sm="MD", comb.fixed=FALSE),
       label.left="Favours A", label.right="Favours B",
       colgap.studlab = grid::unit(2, "cm"),
       colgap.forest.left = grid::unit(0.2, "cm"))
#
# Use slightly different layout for confidence intervals
# (especially useful if upper confidence limit can be negative)
#
settings.meta(CIseparator=" - ")
forest(metagen(-(1:3), (2:4)/10, sm="MD", comb.fixed=FALSE),
       label.left="Favours A", label.right="Favours B",
       colgap.studlab = grid::unit(2, "cm"),
       colgap.forest.left = grid::unit(0.2, "cm"))

# Use old settings
#
settings.meta(oldset)
```

`smoking` *Smoking example*

Description

Meta-analyses on the effect of smoking on mortality risk.

Data have been reconstructed based on the famous Smoking and Health Report to the Surgeon General (Bayne-Jones S et al., 1964). Data sets can be used to evaluate the risk of smoking on overall mortality and lung-cancer deaths, respectively. The person time is attributed such that the rate ratios are equal to the reported mortality ratios implicitely assuming that the data have arisen from a homogeneous age group; more detailed information by age is not available from the report. Note, the group of "non-smokers" actually consists of all participants except those who are smokers of cigarettes only. Information on real non-smokers is not available from the published Smoking and Health Report.

Usage

```
data(smoking)

data(lungcancer)
```

Format

A data frame with the following columns:

study Study label

participants Total number of participants

d.smokers Number of deaths in smokers' group

py.smokers Person years at risk in smokers' group

d.nonsmokers Number of deaths in non-smokers' group

py.nonsmokers Person years at risk in non-smokers' group

Source

Bayne-Jones S et al. (1964), Smoking and Health: Report of the Advisory Committee to the Surgeon General of the United States. U-23 Department of Health, Education, and Welfare. Public Health Service Publication No. 1103. http://profiles.nlm.nih.gov/ps/retrieve/ResourceMetadata/NNBBMQ

See Also

metainc

Examples

```
data(smoking)

m1 <- metainc(d.smokers, py.smokers,
              d.nonsmokers, py.nonsmokers,
              data=smoking, studlab=study)
print(m1, digits=2)

data(lungcancer)

m2 <- metainc(d.smokers, py.smokers,
              d.nonsmokers, py.nonsmokers,
              data=lungcancer, studlab=study)
print(m2, digits=2)
```

---

trimfill.meta *Trim-and-fill method to adjust for bias in meta-analysis*

Description

Trim-and-fill method for estimating and adjusting for the number and outcomes of missing studies in a meta-analysis.

Usage

```
trimfill(x, ...)

## Default S3 method:
trimfill(x, seTE, left=NULL, ma.fixed=TRUE, type="L", n.iter.max=50,
         sm="", studlab=NULL, level=0.95, level.comb=level,
         comb.fixed=FALSE, comb.random=TRUE,
         hakn=FALSE, method.tau="DL",
         prediction=FALSE, level.predict=level,
```

```
          backtransf=TRUE, pscale=1,
          irscale = 1, irunit = "person-years",
          silent=TRUE, ...)

## S3 method for class 'meta'
trimfill(x, left=NULL, ma.fixed=TRUE, type="L", n.iter.max=50,
          level=x$level, level.comb=x$level.comb,
          comb.fixed=FALSE, comb.random=TRUE,
          hakn=x$hakn, method.tau=x$method.tau,
          prediction=x$prediction, level.predict=x$level.predict,
          backtransf=x$backtransf, pscale=x$pscale,
          irscale = x$irscale, irunit = x$irunit,
          silent=TRUE, ...)
```

Arguments

x	An object of class meta, or estimated treatment effect in individual studies.
seTE	Standard error of estimated treatment effect.
left	A logical indicating whether studies are supposed to be missing on the left or right side of the funnel plot. If NULL, the linear regression test for funnel plot symmetry (i.e., function metabias(..., method="linreg")) is used to determine whether studies are missing on the left or right side.
ma.fixed	A logical indicating whether a fixed effect or random effects model is used to estimate the number of missing studies.
type	A character indicating which method is used to estimate the number of missing studies. Either "L" or "R".
n.iter.max	Maximum number of iterations to estimate number of missing studies.
sm	An optional character string indicating underlying summary measure, e.g., "RD", "RR", "OR", "ASD", "HR", "MD", "SMD", or "ROM"; ignored if x is of class meta.
studlab	An optional vector with study labels; ignored if x is of class meta.
level	The level used to calculate confidence intervals for individual studies. If existing, x$level is used as value for level; otherwise 0.95 is used.
level.comb	The level used to calculate confidence interval for the pooled estimate. If existing, x$level.comb is used as value for level.comb; otherwise 0.95 is used.
comb.fixed	A logical indicating whether a fixed effect meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
hakn	A logical indicating whether the method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
method.tau	A character string indicating which method is used to estimate the between-study variance τ^2. Either "DL", "PM", "REML", "ML", "HS", "SJ", "HE", or "EB", can be abbreviated.
prediction	A logical indicating whether a prediction interval should be printed.
level.predict	The level used to calculate prediction interval for a new study.

backtransf	A logical indicating whether results should be back transformed in printouts and plots. If backtransf=TRUE, results for sm="OR" are printed as odds ratios rather than log odds ratios and results for sm="ZCOR" are printed as correlations rather than Fisher's z transformed correlations, for example.
pscale	A numeric giving scaling factor for printing of single event probabilities or risk differences, i.e. if argument sm is equal to "PLOGIT", "PLN", "PRAW", "PAS", "PFT", or "RD".
irscale	A numeric defining a scaling factor for printing of single incidence rates or incidence rate differences, i.e. if argument sm is equal to "IR", "IRLN", "IRS", "IRFT", or "IRD".
irunit	A character specifying the time unit used to calculate rates, e.g. person-years.
silent	A logical indicating whether basic information on iterations shown.
...	other arguments

Details

The trim-and-fill method (Duval, Tweedie 2000a, 2000b) can be used for estimating and adjusting for the number and outcomes of missing studies in a meta-analysis. The method relies on scrutiny of one side of a funnel plot for asymmetry assumed due to publication bias.

Three different methods have been proposed originally to estimate the number of missing studies. Two of these methods (L- and R-estimator) have been shown to perform better in simulations, and are available in this R function (argument type).

A fixed effect or random effects model can be used to estimate the number of missing studies (argument ma.fixed). Furthermore, a fixed effect and/or random effects model can be used to summaries study results (arguments comb.fixed and comb.random). Simulation results (Peters et al. 2007) indicate that the fixed-random model, i.e. using a fixed effect model to estimate the number of missing studies and a random effects model to summaries results, (i) performs better than the fixed-fixed model, and (ii) performs no worse than and marginally better in certain situations than the random-random model. Accordingly, the fixed-random model is the default.

An empirical comparison of the trim-and-fill method and the Copas selection model (Schwarzer et al. 2010) indicates that the trim-and-fill method leads to excessively conservative inference in practice. The Copas selection model is available in R package **metasens**.

The function metagen is called internally.

Value

An object of class c("metagen", "meta", "trimfill"). The object is a list containing the following components:

studlab, sm, left, ma.fixed, type,

n.iter.max, level, level.comb, level.predict,
: As defined above.

comb.fixed, comb.random, prediction, hakn, method.tau

TE, seTE
: Estimated treatment effect and standard error of individual studies.

lower, upper	Lower and upper confidence interval limits for individual studies.
zval, pval	z-value and p-value for test of treatment effect for individual studies.
w.fixed, w.random	Weight of individual studies (in fixed and random effects model).
TE.fixed, seTE.fixed	Estimated overall treatment effect and standard error (fixed effect model).
TE.random, seTE.random	Estimated overall treatment effect and standard error (random effects model).
seTE.predict	Standard error utilised for prediction interval.
lower.predict, upper.predict	Lower and upper limits of prediction interval.
k	Number of studies combined in meta-analysis.
Q	Heterogeneity statistic Q.
tau	Square-root of between-study variance.
method	Pooling method: "Inverse".
call	Function call.
n.iter	Actual number of iterations to estimate number of missing studies.
trimfill	A logical vector indicating studies that have been added by trim-and-fill method.
df.hakn	Degrees of freedom for test of treatment effect for Hartung-Knapp method (only if hakn=TRUE).
title	Title of meta-analysis / systematic review.
complab	Comparison label.
outclab	Outcome label.
label.e	Label for experimental group.
label.c	Label for control group.
label.left	Graph label on left side of forest plot.
label.right	Graph label on right side of forest plot.
k0	Number of studies added by trim-and-fill.
n.e	Number of observations in experimental group (only for object x of class metabin or metacont).
n.c	Number of observations in control group (only for object x of class metabin or metacont).
event.e	Number of events in experimental group (only for object x of class metabin).
event.c	Number of events in control group (only for object x of class metabin).
mean.e	Estimated mean in experimental group (only for object x of class metacont).
sd.e	Standard deviation in experimental group (only for object x of class metacont).
mean.c	Estimated mean in control group (only for object x of class metacont).
sd.c	Standard deviation in control group (only for object x of class metacont).
n	Number of observations (only for object x of class metaprop).
event	Number of events (only for object x of class metaprop).
cor	Corelation (only for object x of class metacor).
class.x	Main class of object x (e.g. 'metabin' or 'metacont').
version	Version of R package **meta** used to create object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Duval S & Tweedie R (2000a), 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 & Tweedie R (2000b), Trim and Fill: A simple funnel-plot-based method of testing and adjusting for publication bias in meta-analysis. *Biometrics*, **56**, 455–463.

Peters JL, Sutton AJ, Jones DR, Abrams KR, Rushton L (2007), Performance of the trim and fill method in the presence of publication bias and between-study heterogeneity. *Statisics in Medicine*, **10**, 4544–62.

Schwarzer G, Carpenter J, Rücker G (2010), Empirical evaluation suggests Copas selection model preferable to trim-and-fill method for selection bias in meta-analysis. *Journal of Clinical Epidemiology*, **63**, 282–8.

See Also

metagen, metabias, funnel

Examples

```
data(Fleiss93)
meta1 <- metabin(event.e, n.e, event.c, n.c,
                 data=Fleiss93, sm="OR")
tf1 <- trimfill(meta1)
summary(tf1)
funnel(tf1)
funnel(tf1, pch=ifelse(tf1$trimfill, 1, 16),
       level=0.9, comb.random=FALSE)
#
# Use log odds ratios on x-axis
#
funnel(tf1, backtransf=FALSE)
funnel(tf1, pch=ifelse(tf1$trimfill, 1, 16),
       level=0.9, comb.random=FALSE, backtransf=FALSE)

trimfill(meta1$TE, meta1$seTE, sm=meta1$sm)
```

---

update.meta *Update a meta-analysis object*

Description

Update an existing meta-analysis object.

Usage

```
## S3 method for class 'meta'
update(object,
       data=object$data, subset=object$subset,
       studlab=object$data$.studlab,
       exclude=object$data$.exclude,
       method=object$method, sm=object$sm,
       incr, allincr=object$allincr,
       addincr=object$addincr, allstudies=object$allstudies,
       MH.exact=object$MH.exact, RR.cochrane=object$RR.cochrane,
       model.glmm = object$model.glmm,
       level=object$level, level.comb=object$level.comb,
       comb.fixed=object$comb.fixed, comb.random=object$comb.random,
       hakn=object$hakn, method.tau=object$method.tau,
       tau.preset=object$tau.preset,
       TE.tau=object$TE.tau, tau.common=object$tau.common,
       prediction=object$prediction, level.predict=object$level.predict,
       null.effect=object$null.effect,
       method.bias=object$method.bias, backtransf = object$backtransf,
       pscale = object$pscale,
       irscale = object$irscale, irunit = object$irunit,
       title=object$title, complab=object$complab, outclab=object$outclab,
       label.e=object$label.e, label.c=object$label.c,
       label.left=object$label.left, label.right=object$label.right,
       n.e=object$n.e, n.c=object$n.c,
       pooledvar=object$pooledvar, method.smd=object$method.smd,
       sd.glass=object$sd.glass, exact.smd=object$exact.smd,
       method.ci=object$method.ci,
       byvar=object$byvar, bylab=object$bylab, print.byvar=object$print.byvar,
       byseparator = object$byseparator,
       print.CMH=object$print.CMH, keepdata=TRUE,
       left=object$left, ma.fixed=object$ma.fixed,
       type=object$type, n.iter.max=object$n.iter.max,
       warn=object$warn, control=object$control, ...)
```

Arguments

object	An object of class meta.
data	Dataset.
subset	Subset.
studlab	Study label.
exclude	An optional vector specifying studies to exclude from meta-analysis, however, to include in printouts and forest plots.
method	A character string indicating which method is to be used for pooling of studies; see metabin and metainc function for admissible values.
sm	A character string indicating which summary measure is used for pooling.

incr	Either a numerical value or vector which can be added to each cell frequency for studies with a zero cell count or the character string "TA" which stands for treatment arm continuity correction.
allincr	A logical indicating if incr is added to each cell frequency of all studies if at least one study has a zero cell count. If FALSE (default), incr is added only to each cell frequency of studies with a zero cell count.
addincr	A logical indicating if incr is added to each cell frequency of all studies irrespective of zero cell counts.
allstudies	A logical indicating if studies with zero or all events in both groups are to be included in the meta-analysis (applies only if sm is equal to "RR" or "OR").
MH.exact	A logical indicating if incr is not to be added to all cell frequencies for studies with a zero cell count to calculate the pooled estimate based on the Mantel-Haenszel method.
RR.cochrane	A logical indicating if 2*incr instead of 1*incr is to be added to n.e and n.c in the calculation of the risk ratio (i.e., sm="RR") for studies with a zero cell. This is used in RevMan 5, the Cochrane Collaboration's program for preparing and maintaining Cochrane reviews.
model.glmm	A character string indicating which GLMM model should be used.
level	The level used to calculate confidence intervals for individual studies.
level.comb	The level used to calculate confidence intervals for pooled estimates.
comb.fixed	A logical indicating whether a fixed effect meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
hakn	A logical indicating whether the method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
method.tau	A character string indicating which method is used to estimate the between-study variance τ^2. Either "DL", "PM", "REML", "ML", "HS", "SJ", "HE", or "EB", can be abbreviated. See function metagen.
tau.preset	Prespecified value for the square-root of the between-study variance τ^2.
TE.tau	Overall treatment effect used to estimate the between-study variance τ^2.
tau.common	A logical indicating whether tau-squared should be the same across subgroups.
prediction	A logical indicating whether a prediction interval should be printed.
level.predict	The level used to calculate prediction interval for a new study.
null.effect	A numeric value specifying the effect under the null hypothesis.
method.bias	A character string indicating which test for funnel plot asymmetry is to be used. Either "rank", "linreg", "mm", "count", "score", or "peters", can be abbreviated. See function metabias
backtransf	A logical indicating whether results should be back transformed in printouts and plots. If backtransf=TRUE, results for sm="OR" are printed as odds ratios rather than log odds ratios and results for sm="ZCOR" are printed as correlations rather than Fisher's z transformed correlations, for example.

pscale	A numeric giving scaling factor for printing of single event probabilities or risk differences, i.e. if argument sm is equal to "PLOGIT", "PLN", "PRAW", "PAS", "PFT", or "RD".
irscale	A numeric defining a scaling factor for printing of single incidence rates or incidence rate differences, i.e. if argument sm is equal to "IR", "IRLN", "IRS", "IRFT", or "IRD".
irunit	A character specifying the time unit used to calculate rates, e.g. person-years.
title	Title of meta-analysis / systematic review.
complab	Comparison label.
outclab	Outcome label.
label.e	Label for experimental group.
label.c	Label for control group.
label.left	Graph label on left side of forest plot.
label.right	Graph label on right side of forest plot.
n.e	Number of observations in experimental group. (only for metagen object)
n.c	Number of observations in control group. (only for metagen object)
pooledvar	A logical indicating if a pooled variance should be used for the mean difference (only for metacont object with sm="MD").
method.smd	A character string indicating which method is used to estimate the standardised mean difference (only for metacont object with sm="SMD"). Either "Hedges" for Hedges' g (default), "Cohen" for Cohen's d, or "Glass" for Glass' delta, can be abbreviated.
sd.glass	A character string indicating which standard deviation is used in the denominator for Glass' method to estimate the standardised mean difference (only for metacont object with sm="SMD"). Either "control" using the standard deviation in the control group (sd.c) or "experimental" using the standard deviation in the experimental group (sd.e), can be abbreviated.
exact.smd	A logical indicating whether exact formulae should be used in estimation of the standardised mean difference and its standard error.
method.ci	A character string indicating which method is used to calculate confidence intervals for individual studies. Either "CP", "WS", "WSCC", "AC", "SA",, "SACC", or "NAsm", can be abbreviated. See function metaprop.
byvar	An optional vector containing grouping information (must be of same length as event.e).
bylab	A character string with a label for the grouping variable.
print.byvar	A logical indicating whether the name of the grouping variable should be printed in front of the group labels.
byseparator	A character string defining the separator between label and levels of grouping variable.
print.CMH	A logical indicating whether result of the Cochran-Mantel-Haenszel test for overall effect should be printed.
keepdata	A logical indicating whether original data (set) should be kept in meta object.

left	A logical indicating whether studies are supposed to be missing on the left or right side of the funnel plot. If NULL, the linear regression test for funnel plot symmetry (i.e., function metabias(..., method="linreg")) is used to determine whether studies are missing on the left or right side.
ma.fixed	A logical indicating whether a fixed effect or random effects model is used to estimate the number of missing studies.
type	A character indicating which method is used to estimate the number of missing studies. Either "L" or "R".
n.iter.max	Maximum number of iterations to estimate number of missing studies.
warn	A logical indicating whether warnings should be printed (e.g., if incr is added to studies with zero cell frequencies).
control	An optional list to control the iterative process to estimate the between-study variance tau^2. This argument is passed on to rma.uni or rma.glmm, respectively.
...	Additional arguments (ignored at the moment).

Details

Wrapper function to update an existing meta-analysis object which was created with R function metabin, metacont, metacor, metagen, metainc, metamean, metaprop, or metarate. More details on function arguments are available in help files of respective R functions

This function can also be used for objects of class 'trimfill', 'metacum', and 'metainf'.

Value

An object of class "meta" and "metabin", "metacont", "metacor", "metainc", "metagen", "metamean", "metaprop", or "metarate".

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

metabin, metacont, metacor, metagen, metainc, metamean, metaprop, metarate

Examples

```
data(Fleiss93cont)
meta1 <- metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c,
                  data=Fleiss93cont, sm="SMD", studlab=study)
meta1

# Change summary measure (from 'SMD' to 'MD')
#
update(meta1, sm="MD")

# Restrict analysis to subset of studies
```

```
#
update(meta1, subset=1:2)

# Use different levels for confidence intervals
#
meta2 <- update(meta1, level=0.66, level.comb=0.99)
print(meta2, digits=2)
forest(meta2)
```

`weights.meta` *Calculate absolute and percentage weights for meta-analysis*

Description

The `weights.meta` method returns a data frame containing information on absolute and percentage weights of individual studies contributing to fixed effect and random effects meta-analysis.

Usage

```
## S3 method for class 'meta'
weights(object, comb.fixed = object$comb.fixed,
        comb.random = object$comb.random, ...)
```

Arguments

`object`	An object of class `meta`.
`comb.fixed`	A logical indicating whether absolute and percentage weights from the fixed effect model should be calculated.
`comb.random`	A logical indicating whether absolute and percentage weights from the random effects model should be calculated.
`...`	Additional arguments (ignored at the moment).

Value

A data frame with the following variables is returned:

Variable	Definition	Condition
w.fixed	absolute weights in fixed effect model	(if `comb.fixed = TRUE`)
p.fixed	percentage weights in fixed effect model	(if `comb.fixed = TRUE`)
w.random	absolute weights in random effects model	(if `comb.random = TRUE`)
p.random	percentage weights in random effects model	(if `comb.random = TRUE`)

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

metabin, metacont, metagen

Examples

```
data(Fleiss93cont)
#
# Do meta-analysis (fixed effect and random effects model)
#
meta1 <- metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c, study,
                  data=Fleiss93cont, sm="SMD")
#
# Print weights for fixed effect and random effects meta-analysis
#
weights(meta1)

#
# Do meta-analysis (only random effects model)
#
meta2 <- update(meta1, comb.fixed = FALSE)
#
# Print weights for random effects meta-analysis
#
weights(meta2)
#
# Print weights for fixed effect and random effects meta-analysis
#
weights(meta2, comb.fixed = TRUE)
```

---

woodyplants *Elevated CO_2 and total biomass of woody plants*

Description

Meta-analysis on effects of elevated CO_2 on total biomass of woody plants

Usage

```
data(woodyplants)
```

Format

A data frame with the following columns:

obsno Observation number

papno Database paper number

treat Treatment code

level Treatment level

n.elev Number of observations in experimental group (elevated CO_2-level)

mean.elev Estimated mean in experimental group

sd.elev Standard deviation in experimental group

n.amb Number of observations in control group (ambient CO_2-level)

mean.amb Estimated mean in control group

sd.amb Standard deviation in control group

Details

This dataset has been used as an example in Hedges et al. (1999) to describe methods for the meta-analysis of response ratios. The complete dataset with 102 observations and 26 variables is available online as a supplement. Here only a subset of 10 variables is provided and used in the examples.

Source

Website http://www.esapubs.org/archive/ecol/E080/008/

References

Hedges LV, Gurevitch J, Curtis PS (1999), The meta-analysis of response ratios in experimental ecology. *Ecology*, **80**, 1150–6.

Examples

```
data(woodyplants)

# Meta-analysis of response ratios (Hedges et al., 1999)
#
meta8 <- metacont(n.elev, mean.elev, sd.elev,
                  n.amb, mean.amb, sd.amb,
                  data=woodyplants, sm="ROM",
                  studlab=paste(obsno, papno, sep = " / "))
summary(meta8, prediction=TRUE)

# Meta-analysis for plants grown with low soil fertility treatment
#
meta9 <- update(meta8, subset=(treat=="fert"&level=="low"))
summary(meta9, prediction=TRUE)

# Meta-analysis for plants grown under low light conditions
#
meta10 <- update(meta8, subset=(treat=="light"&level=="low"))
summary(meta10, prediction=TRUE)
```

Index