

# An Overview of Functions in the *metafor* Package

last updated: Nov 8 2021  
(not all functions documented)

given the required data (e.g., means, SDs, and group sizes; counts for 2x2 tables; correlations and sample sizes), calculate the desired effect size or outcome measure for the meta-analysis (e.g., raw or standardized mean differences, log odds ratios, log risk ratios, risk differences, r-to-z transformed correlations, ...)

functions in the 'util' package to:

- read in data from ASCII file
- see also 'foreign', 'readxl', and 'haven' packages for reading in other data formats

**read.table()**  
**read.csv()**  
**read.delim()**

**escalc()**

- yi = observed outcomes or effect size estimates
- vi = corresponding sampling variances

**rma.uni()**  
**rma.mh()**  
**rma.peto()**  
**rma.glmm()**  
**rma.mv()**

- rma.uni() = equal/fixed- and random/mixed-effects models ("inverse-variance" method; normal-normal models)
- rma.mh() = Mantel-Haenszel method
- rma.peto() = Peto's method (equal/fixed-effects model)
- rma.glmm() = equal/fixed- and random/mixed-effects models (binomial-normal and Poisson-normal models)
- rma.mv() = equal/fixed- and random/mixed-effects multivariate/multilevel models (normal-normal models)

note: rma.uni() takes either 'yi' and 'vi' as input or one can supply the required data to calculate the desired effect size or outcome measure for the meta-analysis directly; rma.mh(), rma.peto(), and rma.glmm() require that the raw counts are supplied; rma.mv() takes 'yi' and 'V' as input (V is the variance-covariance matrix of the sampling errors)

**print()**  
**summary()**  
**aggregate()**

*print functions*

*fitted and predicted values*

*residuals and influential case diagnostics*

*funnel plot asymmetry / publication bias*

*confidence intervals and inference*

*plotting functions*

*various extractor functions*

**print()**  
**summary()**

**fitted()**  
**predict()**  
**blup()**  
**ranef()**  
**cumul()**

**residuals()**  
**rstandard()**  
**rstudent()**  
**hatvalues()**  
**weights()**  
**influence()**  
**leave1out()**

**ranktest()**  
**regtest()**  
**trimfill()**  
**hc()**  
**tes()**  
**selmodel()**

**confint()**  
**anova()**  
**permutest()**  
**robust()**  
**vif()**

**forest()**  
**funnel()**  
**labbe()**  
**radial()**  
**qqnorm()**  
**baujat()**  
**gosh()**  
**regplot()**  
**plot()**

**logLik()**  
**deviance()**  
**fitstats()**  
**AIC(), BIC()**  
**coef()**  
**vcov()**

note: class of fitted model object is the same as the function name; so print() for an object of class 'rma.uni' actually calls print.rma.uni() and so on

note: blup() only for 'rma.uni' objects; ranef() only for 'rma.uni' and 'rma.mv' objects; cumul() not for 'rma.mv' or 'rma.glmm' objects

note: all functions implemented for 'rma.uni' objects; coverage of functions for other objects varies (see docs)

note: regtest() not for 'rma.glmm' or 'rma.mv' objects; trimfill(), hc(), tes(), selmodel() only for 'rma.uni' objects

note: confint() not for 'rma.glmm' objects; anova() and robust() only for 'rma.uni' and 'rma.mv' objects; permutest() only for 'rma.uni' objects

note: forest() and funnel() also take 'yi' and 'vi' as input; qqnorm(), baujat(), gosh() and plot() not for 'rma.glmm' or 'rma.mv' objects

note: coef() also for 'permutest.rma.uni' and 'summary.rma' objects