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’ package for reading in other data formats

• escalc() = observed outcomes or effect size estimates
• vi = corresponding sampling variances

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

note: regtest() not for ‘rma.glmm’ or ‘rma.mv’ objects; trimfill() and hc() only for ‘rma.uni’ objects
note: coef() also for ‘permutest.rma.uni’ and ‘summary.rma’ objects

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 is more limited (see docs)

note: confint() not for ‘rma.glmm’ or ‘rma.mv’ objects; trimfill() and hc() only for ‘rma.uni’ objects
note: coef() also 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 ‘V’ as input; qqnorm(), baujat(), gosh() and plot() not for ‘rma.glmm’ or ‘rma.mv’ objects

• residuals() = residuals and influential case diagnostics
• fitted() = fitted and predicted values
• predict() = predict functions
• rstandard() = print functions
• rstudent() = plot functions
• hatvalues() = various extractor functions
• weights() = funnel plot symmetry (publication bias)
• influence() = confidence intervals and inference
• leave1out() = 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

• ranktest() = funnel plot asymmetry (publication bias)
• confint() = 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

• forest() = note: forest() and funnel() also take ‘yi’ and ‘V’ as input; qqnorm(), baujat(), gosh() and plot() not for ‘rma.glmm’ or ‘rma.mv’ objects

• logLik() = note: coef() also for ‘rma.glmm’ objects
• deviance() = note: coef() also for ‘rma.glmm’ objects
• AIC(), BIC() = note: coef() also for ‘rma.glmm’ objects
• anova() = note: coef() also for ‘rma.glmm’ objects
• robust() = note: coef() also for ‘rma.glmm’ objects
• permutest() = note: coef() also for ‘rma.glmm’ objects
• coef() = note: coef() also for ‘rma.glmm’ objects

last updated: May 09 2018