The metafor package in R provides a wide range of functions for conducting meta-analyses. Here is a summary of some of its key functions, along with brief descriptions:

- **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 binomial-normal and Poisson-normal models
- **rma.mv()**: Equal/fixed- and random/mixed-effects multivariate/multilevel models (normal-normal models)

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

---

### An Overview of Functions in the metafor Package

(last updated: Febv 26 2023
(not all functions documented)

---

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>rma.uni()</code></td>
<td>Equal/fixed- and random/mixed-effects models (&quot;inverse-variance&quot; method; normal-normal models)</td>
</tr>
<tr>
<td><code>rma.mh()</code></td>
<td>Mantel-Haenszel method</td>
</tr>
<tr>
<td><code>rma.peto()</code></td>
<td>Peto's method (equal/fixed-effects model)</td>
</tr>
<tr>
<td><code>rma.glmm()</code></td>
<td>Equal/fixed- and random/mixed-effects binomial-normal and Poisson-normal models</td>
</tr>
<tr>
<td><code>rma.mv()</code></td>
<td>Equal/fixed- and random/mixed-effects multivariate/multilevel models (normal-normal models)</td>
</tr>
</tbody>
</table>

### Functions for reading data formats
- `read.table()`
- `read.csv()`
- `read.delim()`

### Functions for calculating effect sizes and variances
- `escalc()`
- `vcalc()`
- `rcalc()`
- `conv.wald()`
- `conv.fivenum()`
- `conv.2x2()`
- `conv.delta()`

### Functions for fitted models
- `fitted()`
- `predict()`
- `blup()`
- `ranef()`
- `cumul()`

### Functions for residuals and influence
- `residuals()`
- `rstandard()`
- `rstudent()`
- `hatvalues()`
- `weights()`
- `influence()`
- `leave1out()`

### Functions for confidence intervals and hypothesis testing
- `confint()`
- `anova()`
- `regtest()`
- `trimfill()`
- `permutest()`
- `robust()`
- `vif()`

### Plotting functions
- `forest()`
- `funnel()`
- `labbe()`
- `radial()`
- `qqnorm()`
- `baujat()`
- `gosh()`
- `regplot()`
- `plot()`

### Other functions
- `logLik()`
- `deviance()`
- `fitstats()`
- `AIC()`, `BIC()`
- `coef()`
- `vcov()`

---

**Note:**
- `rma.uni()` takes either ‘yi’ and ‘vi’ as input; `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()` for an object of class ‘rma.uni’ actually calls `print.rma.uni()` and so on
- `blup()` only for ‘rma.uni’ objects; `ranef()` only for ‘rma.uni’ and ‘rma.mv’ objects; `cumul()` not for ‘rma.mv’ or ‘rma.glmm’ objects
- `regtest()` not for ‘rma.glmm’ objects; `trimfill()`, `hc()`, `tes()`, `selmodel()` only for ‘rma.uni’ objects
- `confint()` not for ‘rma.glmm’ objects; `anova()` and `robust()` only for ‘rma.uni’ and ‘rma.mv’ objects; `permutest()` only for ‘rma.uni’ objects
- `forest()` and `funnel()` also take ‘yi’ and ‘vi’ as input; `qqnorm()`, `baujat()`, `gosh()` and `plot()` not for ‘rma.glmm’ or ‘rma.mv’ objects
- `coef()` also for ‘permutest.rma.uni’ and ‘summary.rma’ objects

---

An overview diagram is also provided, showing the relationships between functions and their inputs and outputs.