Package 'ConfoundedMeta'

August 14, 2017

Type Package

Title Sensitivity Analyses for Unmeasured Confounding in Meta-Analyses

Version 1.3.0

Date 2017-05-31

Author Maya B. Mathur, Tyler J. VanderWeele

Maintainer Maya B. Mathur <maya.z.mathur@gmail.com>

Description Conducts sensitivity analyses for unmeasured confounding in random-effects meta-analysis per Mathur & VanderWeele (in preparation). Given output from a random-effects meta-analysis with a relative risk outcome, computes point estimates and inference for: (1) the proportion of studies with true causal effect sizes more extreme than a specified threshold of scientific significance; and (2) the minimum bias factor and confounding strength required to reduce to less than a specified threshold the proportion of studies with true effect sizes of scientifically significant size. Creates plots and tables for visualizing these metrics across a range of bias values. Provides tools to easily scrape study-level data from a published forest plot or summary table to obtain the needed estimates when these are not reported.

License GPL-2

Imports ggplot2 (>= 2.2.1), metafor, stats

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-08-14 13:17:55 UTC

R topics documented:

confounded_meta *Estimates and inference for sensitivity analyses*

Description

Computes point estimates, standard errors, and confidence interval bounds for (1) `prop`, the proportion of studies with true effect sizes above `.q` (or below `.q` for an apparently preventive `.yr`) as a function of the bias parameters; (2) the minimum bias factor on the relative risk scale (`Tmin`) required to reduce to less than `.r` the proportion of studies with true effect sizes more extreme than `.q`; and (3) the counterpart to (2) in which bias is parameterized as the minimum relative risk for both confounding associations (`Gmin`).

Usage

```
confounded_meta(.q, .r = NULL, .muB = NULL, .sigB = 0, .yr, .vyr = NULL,
  .t2, .vt2 = NULL, CI.level = 0.95, .tail = NULL)
```

Arguments

`.q`	True effect size that is the threshold for "scientific significance"
`.r`	For `Tmin` and `Gmin`, value to which the proportion of large effect sizes is to be reduced
`.muB`	Mean bias factor on the log scale across studies
`.sigB`	Standard deviation of log bias factor across studies
`.yr`	Pooled point estimate (on log scale) from confounded meta-analysis
`.vyr`	Estimated variance of pooled point estimate from confounded meta-analysis
`.t2`	Estimated heterogeneity (tau^2) from confounded meta-analysis
`.vt2`	Estimated variance of tau^2 from confounded meta-analysis
`CI.level`	Confidence level as a proportion
`.tail`	`above` for the proportion of effects above `.q`; `below` for the proportion of effects below `.q`. By default, is set to `above` for relative risks above 1 and to `below` for relative risks below 1.

Details

To compute all three point estimates (`prop, Tmin, and Gmin`) and inference, all arguments must be non-`NULL`. To compute only a point estimate for `prop`, arguments `.r, .vyr`, and `.vt2` can be left `NULL`. To compute only point estimates for `Tmin` and `Gmin`, arguments `.muB, .vyr`, and `.vt2` can be left `NULL`. To compute inference for all point estimates, `.vyr` and `.vt2` must be supplied.

Examples

```
d = metafor::escalc(measure="RR", ai=tpos, bi=tneg,
ci=cpos, di=cneg, data=metafor::dat.bcg)

m = metafor::rma.uni(yi= d$yi, vi=d$vi, knha=FALSE,
                     measure="RR", method="DL" )
yr = as.numeric(m$b)  # metafor returns on log scale
vyr = as.numeric(m$vb)
t2 = m$tau2
vt2 = m$se.tau2^2

# obtaining all three estimators and inference
confounded_meta( .q=log(0.90), .r=0.20, .muB=log(1.5), .sigB=0.1,
                 .yr=yr, .vyr=vyr, .t2=t2, .vt2=vt2,
                 CI.level=0.95 )

# passing only arguments needed for prop point estimate
confounded_meta( .q=log(0.90), .muB=log(1.5),
                 .yr=yr, .t2=t2, CI.level=0.95 )

# passing only arguments needed for Tmin, Gmin point estimates
confounded_meta( .q=log(0.90), .r=0.20,
                 .yr=yr, .t2=t2, CI.level=0.95 )
```

`scrape_meta` *Convert forest plot or summary table to meta-analytic dataset*

Description

Given relative risks (RR) and upper bounds of 95% confidence intervals (CI) from a forest plot or summary table, returns a dataframe ready for meta-analysis (e.g., via the `metafor` package) with the log-RRs and their variances. Optionally, the user may indicate studies for which the point estimate is to be interpreted as an odds ratios of a common outcome rather than a relative risk; for such studies, the function applies VanderWeele (2017)'s square-root transformation to convert the odds ratio to an approximate risk ratio.

Usage

```
scrape_meta(.type = "RR", .est, .hi, .sqrt = FALSE)
```

Arguments

`.type`	`RR` if point estimates are RRs or ORs (to be handled on log scale); `raw` if point estimates are raw differences, standardized mean differences, etc. (such that they can be handled with no transformations)
`.est`	Vector of study point estimates on RR or OR scale
`.hi`	Vector of upper bounds of 95% CIs on RRs

`.sqrt`	Vector of booleans (TRUE/FALSE) for whether each study measured an odds ratio of a common outcome that should be approximated as a risk ratio via the square-root transformation

`sens_plot` *Plots for sensitivity analyses*

Description

Produces line plots (`.type=="line"`) showing the bias factor on the relative risk (RR) scale vs. the proportion of studies with true RRs above `.q` (or below it for an apparently preventive relative risk). The plot secondarily includes a X-axis scaled based on the minimum strength of confounding to produce the given bias factor. The shaded region represents a 95% pointwise confidence band. Alternatively, produces distribution plots (`.type=="dist"`) for a specific bias factor showing the observed and true distributions of RRs with a red line marking exp(`.q`).

Usage

```
sens_plot(.type, .q, .muB = NULL, .Bmin = log(1), .Bmax = log(5),
  .sigB = 0, .yr, .vyr = NULL, .t2, .vt2 = NULL, breaks.x1 = NULL,
  breaks.x2 = NULL, CI.level = 0.95)
```

Arguments

`.type`	`dist` for distribution plot; `line` for line plot (see Details)
`.q`	True effect size that is the threshold for "scientific significance"
`.muB`	Single mean bias factor on log scale (only needed for distribution plot)
`.Bmin`	Lower limit of lower X-axis on the log scale (only needed for line plot)
`.Bmax`	Upper limit of lower X-axis on the log scale (only needed for line plot)
`.sigB`	Standard deviation of log bias factor across studies (length 1)
`.yr`	Pooled point estimate (on log scale) from confounded meta-analysis
`.vyr`	Estimated variance of pooled point estimate from confounded meta-analysis
`.t2`	Estimated heterogeneity (tau^2) from confounded meta-analysis
`.vt2`	Estimated variance of tau^2 from confounded meta-analysis
`breaks.x1`	Breaks for lower X-axis (bias factor) on RR scale
`breaks.x2`	Breaks for upper X-axis (confounding strength) on RR scale
`CI.level`	Poitnwise confidence level as a proportion

Details

Arguments `.vyr` and `.vt2` can be left `NULL`, in which case no confidence band will appear on the line plot.

Examples

```
# with variable bias and with confidence band
sens_plot( .type="line", .q=log(1.1), .Bmin=log(1), .Bmax=log(4), .sigB=0.1,
           .yr=log(1.3), .vyr=0.005, .t2=0.4, .vt2=0.03 )

# with fixed bias and without confidence band
sens_plot( .type="line", .q=log(1.1), .Bmin=log(1), .Bmax=log(4),
           .yr=log(1.3), .t2=0.4 )

# apparently preventive
sens_plot( .type="line", .q=log(0.90), .Bmin=log(1), .Bmax=log(4),
           .yr=log(0.6), .vyr=0.005, .t2=0.4, .vt2=0.04 )

# distribution plot: apparently causative
# commented out because takes 5-10 seconds to run
# sens_plot( .type="dist", .q=log(1.1), .muB=log(2),
#           .yr=log(1.3), .t2=0.4 )

# distribution plot: apparently preventive
# commented out because takes 5-10 seconds to run
# sens_plot( .type="dist", .q=log(0.90), .muB=log(1.5),
#           .yr=log(0.7), .t2=0.2 )
```

`sens_table` *Tables for sensitivity analyses*

Description

Produces table showing the proportion of true effect sizes more extreme than `.q` across a grid of bias parameters `.muB` and `.sigB` (for `.meas == "prop"`). Alternatively, produces a table showing the minimum bias factor (for `.meas == "Tmin"`) or confounding strength (for `.meas == "Gmin"`) required to reduce to less than `.r` the proportion of true effects more extreme than `.q`.

Usage

```
sens_table(.meas, .q, .r = seq(0.1, 0.9, 0.1), .muB = NULL, .sigB = NULL,
  .yr, .t2)
```

Arguments

`.meas`	`prop`, `Tmin`, or `Gmin`
`.q`	True effect size that is the threshold for "scientific significance"
`.r`	For `Tmin` and `Gmin`, vector of values to which the proportion of large effect sizes is to be reduced
`.muB`	Mean bias factor on the log scale across studies
`.sigB`	Standard deviation of log bias factor across studies
`.yr`	Pooled point estimate (on log scale) from confounded meta-analysis
`.t2`	Estimated heterogeneity (tau^2) from confounded meta-analysis

Details

For `.meas=="Tmin"` or `.meas=="Gmin"`, arguments `.muB` and `.sigB` can be left `NULL`; `.r` can also be `NULL` as it will default to a reasonable range of proportions. Returns a `data.frame` whose rows are values of `.muB` (for `.meas=="prop"`) or of `.r` (for `.meas=="Tmin"` or `.meas=="Gmin"`). Its columns are values of `.sigB` (for `.meas=="prop"`) or of `.q` (for `.meas=="Tmin"` or `.meas=="Gmin"`). Tables for `Gmin` will display `NaN` for cells corresponding to `Tmin`<1, i.e., for which no bias is required to reduce the effects as specified.

Examples

```
sens_table( .meas="prop", .q=log(1.1), .muB=c( log(1.1),
log(1.5), log(2.0) ), .sigB=c(0, 0.1, 0.2),
.yr=log(2.5), .t2=0.1 )

sens_table( .meas="Tmin", .q=c( log(1.1), log(1.5) ),
.yr=log(1.3), .t2=0.1 )

# will have NaNs in cells with Tmin < 1 (no bias needed)
sens_table( .meas="Gmin", .r=0.8, .q=c( log(1.1) ),
.yr=log(1.3), .t2=0.1 )
```

`stronger_than` *Estimate proportion of population effect sizes above or below a threshold*

Description

This is a wrapper for `confounded_meta` that estimates, without any adjustment for confounding bias, the proportion of effect sizes above or below a specified threshold. Effect sizes may be of any type (they need not be relative risks).

Usage

```
stronger_than(.q, .yr, .vyr = NULL, .t2, .vt2 = NULL, CI.level = 0.95,
  .tail)
```

Arguments

`.q`	True effect size that is the threshold for "scientific significance"
`.yr`	Pooled point estimate from meta-analysis
`.vyr`	Estimated variance of pooled point estimate from meta-analysis
`.t2`	Estimated heterogeneity (tau^2) from meta-analysis
`.vt2`	Estimated variance of tau^2 from meta-analysis
`CI.level`	Confidence level as a proportion
`.tail`	`above` for the proportion of effects above `.q`; `below` for the proportion of effects below `.q`.

Index