Package 'EValue'

May 18, 2018

Type Package

Title Sensitivity Analyses for Unmeasured Confounding in Observational Studies and Meta-Analyses

Version 1.1.4

Date 2018-4-3

Author Maya B. Mathur, Peng Ding, Tyler J. VanderWeele

Maintainer Maya B. Mathur <mmathur@stanford.edu>

Description Conducts sensitivity analyses for unmeasured confounding for either an observational study or a meta-analysis of observational studies. For a single observational study, the package reports E-values, defined as the minimum strength of association on the risk ratio scale that an unmeasured confounder would need to have with both the treatment and the outcome to fully explain away a specific treatment-outcome association, conditional on the measured covariates. One can use one of the evalues.XX() functions to compute E-values for the relevant outcome types. Outcome types include risk ratios, odds ratio with common or rare outcomes, hazard ratios with common or rare outcomes, and standardized differences in outcomes. Optionally, one can use the bias_plot() function to plot the bias factor as a function of two sensitivity parameters. (See VanderWeele & Ding, 2017 [<http://annals.org/aim/article/2643434>] for details.) For a meta-analysis, use the function confounded_meta to compute point estimates and inference for: (1) the proportion of studies with true causal effect sizes more extreme than a specified threshold of scientific importance; 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. The functions sens_plot() and sens_table() create plots and tables for visualizing these meta-analysis metrics across a range of bias values, and scrape_meta helps scrape study-level data from a published forest plot or summary table to obtain the needed estimates when these are not reported. (See Mathur & VanderWeele [<https://osf.io/jkhfg/>] for details.) Most of the analyses available in this package can also be conducted using web-based graphical interfaces (for a single observational study: <https://mmathur.shinyapps.io/evalue/>; for a meta-analysis: <https://mmathur.shinyapps.io/meta_gui_2/>).

LazyData true

License GPL-2

Imports stats, graphics, ggplot2 (>= 2.2.1), metafor, msm, devtools

RoxygenNote 6.0.1

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-18 21:51:20 UTC

R topics documented:

`bias_plot` *Plot bias factor as function of confounding relative risks*

Description

Plots the bias factor required to explain away a provided relative risk.

Usage

```
bias_plot(RR, xmax)
```

Arguments

`RR`	The relative risk
`xmax`	Upper limit of x-axis.

Examples

```
# recreate the plot in VanderWeele and Ding (2017)
bias_plot(RR=3.9, xmax=20)
```

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 = NA, muB = NA, sigB = 0, yr, vyr = NA, t2,
  vt2 = NA, CI.level = 0.95, tail = NA)
```

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-NA. To compute only a point estimate for prop, arguments r, vyr, and vt2 can be left NA. To compute only point estimates for Tmin and Gmin, arguments muB, vyr, and vt2 can be left NA. 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 )
```

`evalues.HR` *Compute E-value for a hazard ratio and its confidence interval limits*

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit on the risk ratio scale (through an approximate conversion if needed when outcome is common) as well as E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
evalues.HR(est, lo = NA, hi = NA, rare = NA, true = 1)
```

Arguments

`est`	The point estimate
`lo`	The lower limit of the confidence interval
`hi`	The upper limit of the confidence interval
`rare`	1 if outcome is rare (<15 percent at end of follow-up); 0 if outcome is not rare (>15 percent at end of follow-up)
`true`	The true HR to which to shift the observed point estimate. Typically set to 1 to consider a null true effect.

Examples

```
# compute E-value for HR = 0.56 with CI: [0.46, 0.69]
# for a common outcome
evalues.HR( 0.56, 0.46, 0.69, rare = FALSE )
```

`evalues.MD` *Compute E-value for a difference of means and its confidence interval limits*

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit on the risk ratio scale (through an approximate conversion) as well as E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
evalues.MD(est, se = NA, true = 0)
```

Arguments

`est`	The point estimate as a standardized difference (i.e., Cohen's d)
`se`	The standard error of the point estimate
`true`	The true standardized mean difference to which to shift the observed point estimate. Typically set to 0 to consider a null true effect.

Examples

```
# compute E-value if Cohen's d = 0.5 with SE = 0.25
evalues.MD( .5, .25 )
```

`evalues.OR` *Compute E-value for an odds ratio and its confidence interval limits*

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit on the risk ratio scale (through an approximate conversion if needed when outcome is common) as well as E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
evalues.OR(est, lo = NA, hi = NA, rare = NA, true = 1)
```

Arguments

est	The point estimate
lo	The lower limit of the confidence interval
hi	The upper limit of the confidence interval
rare	1 if outcome is rare (<15 percent at end of follow-up); 0 if outcome is not rare (>15 percent at end of follow-up)
true	The true OR to which to shift the observed point estimate. Typically set to 1 to consider a null true effect.

Examples

```
# compute E-values for OR = 0.86 with CI: [0.75, 0.99]
# for a common outcome
evalues.OR( 0.86, 0.75, 0.99, rare = FALSE )

## Example 2
## Hsu and Small (2013 Biometrics) Data
## sensitivity analysis after log-linear or logistic regression

head(lead)

## log linear model -- obtain the conditional risk ratio
lead.loglinear = glm(lead ~ ., family = binomial(link = "log"),
                         data = lead)
est = summary( lead.loglinear )$coef[2, c(1, 2)]

RR       = exp(est[1])
lowerRR  = exp(est[1] - 1.96*est[2])
upperRR  = exp(est[1] + 1.96*est[2])
evalues.RR(RR, lowerRR, upperRR)

## logistic regression -- obtain the conditional odds ratio
lead.logistic = glm(lead ~ ., family = binomial(link = "logit"),
                        data = lead)
est = summary( lead.logistic )$coef[2, c(1, 2)]

OR       = exp(est[1])
lowerOR  = exp(est[1] - 1.96*est[2])
upperOR  = exp(est[1] + 1.96*est[2])
evalues.OR(OR, lowerOR, upperOR, rare=FALSE)
```

`evalues.RD` *Compute E-value for a population-standardized risk difference and its confidence interval limits*

Description

Returns E-values for the point estimate and the lower confidence interval limit for a positive risk difference. If the risk difference is negative, the exposure coding should be first be reversed to yield a positive risk difference.

Usage

```
evalues.RD(n11, n10, n01, n00, true = 0, alpha = 0.05, grid = 1e-04)
```

Arguments

n11	Number of exposed, diseased individuals
n10	Number of exposed, non-diseased individuals
n01	Number of unexposed, diseased individuals
n00	Number of unexposed, non-diseased individuals
true	True value of risk difference to which to shift the point estimate. Usually set to 0 to consider the null.
alpha	Alpha level
grid	Spacing for grid search of E-value

Examples

```
## example 1
## Hammond and Holl (1958 JAMA) Data
## Two by Two Table
##          Lung Cancer    No Lung Cancer
##Smoker    397            78557
##Nonsmoker 51             108778

# E-value to shift observed risk difference to 0
evalues.RD( 397, 78557, 51, 108778)

# E-values to shift observed risk difference to other null values
evalues.RD( 397, 78557, 51, 108778, true = 0.001)
```

`evalues.RR` *Compute E-value for a risk ratio or rate ratio and its confidence interval limits*

Description

Returns a data frame containing point estimates, the lower confidence limit, and the upper confidence limit for the risk ratio (as provided by the user) as well as E-values for the point estimate and the confidence interval limit closer to the null.

Usage

```
evalues.RR(est, lo = NA, hi = NA, true = 1)
```

Arguments

est	The point estimate
lo	The lower limit of the confidence interval
hi	The upper limit of the confidence interval
true	The true RR to which to shift the observed point estimate. Typically set to 1 to consider a null true effect.

Examples

```
# compute E-value for leukemia example in VanderWeele and Ding (2017)
evalues.RR( 0.80, 0.71, 0.91 )

# you can also pass just the point estimate
evalues.RR( 0.80 )

# demonstrate symmetry of E-value
# this apparently causative association has same E-value as the above
evalues.RR( 1 / 0.80 )
```

---

lead *An example dataset*

Description

An example dataset from Hsu and Small (Biometrics, 2013).

Usage

```
lead
```

Format

An object of class `data.frame` with 3340 rows and 18 columns.

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 = NA, Bmin = log(1), Bmax = log(5), sigB = 0, yr,
  vyr = NA, t2, vt2 = NA, breaks.x1 = NA, breaks.x2 = NA,
  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 (optional for line plot; not used for distribution plot)
breaks.x2	Breaks for upper X-axis (confounding strength) on RR scale (optional for line plot; not used for distribution plot)
CI.level	Pointwise confidence level as a proportion

Details

Arguments vyr and vt2 can be left NA, 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 = NA, sigB = NA, 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 NA; r can also be NA 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 )

# Tmin is 1 here because we already have <80% of effects
#  below log(1.1) even without any confounding
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

Estimates the proportion of true effect sizes in a meta-analysis above or below a specified threshold of scientific importance. Effect sizes may be of any type (they need not be relative risks). This is a wrapper for `confounded_meta`; it is the special case in which there is no unmeasured confounding.

Usage

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

Arguments

`q`	True effect size that is the threshold for "scientific importance"
`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`.

threshold *Compute E-value for single value of risk ratio*

Description

Computes E-value for a single value of the risk ratio. Users should typically call the relevant `evalues.XX()` function rather than this internal function.

Usage

```
threshold(x, true = 1)
```

Arguments

`x`	The risk ratio
`true`	The true RR to which to shift the observed point estimate. Typically set to 1 to consider a null true effect.

Examples

```
## Example 1
## Hammond and Holl (1958 JAMA) Data
## Two by Two Table
#          Lung Cancer    No Lung Cancer
# Smoker    397            78557
# Nonsmoker 51             108778

# first get RR and CI bounds
twoXtwoRR(397, 78557, 51, 108778)

# then compute E-values
evalues.RR(10.729780, 8.017457, 14.359688)
```

`twoXtwoRR` *Estimate risk ratio and compute CI limits from two-by-two table*

Description

Given counts in a two-by-two table, computes risk ratio and confidence interval limits.

Usage

```
twoXtwoRR(n11, n10, n01, n00, alpha = 0.05)
```

Arguments

`n11`	Number exposed (X=1) and diseased (D=1)
`n10`	Number exposed (X=1) and not diseased (D=0)
`n01`	Number unexposed (X=0) and diseased (D=1)
`n00`	Number unexposed (X=0) and not diseased (D=0)
`alpha`	Alpha level associated with confidence interval

Examples

```
# Hammond and Holl (1958 JAMA) Data
# Two by Two Table
#          Lung Cancer    No Lung Cancer
# Smoker    397            78557
# Nonsmoker 51             108778

twoXtwoRR(397, 78557, 51, 108778)
```

Index