Package 'MetaUtility'

April 10, 2019

Type Package

Title Utility Functions for Conducting and Reporting Meta-Analysis

Version 1.2.0

Author Maya B. Mathur, Tyler J. VanderWeele

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

Description Contains functions to estimate the proportion of effects stronger than a threshold of scientific importance (per Mathur & VanderWeele, 2018 [<https://onlinelibrary.wiley.com/doi/full/10.1002/sim.8057>]), to make various effect size conversions, to compute and format inference in a meta-analysis, and to scrape results from existing meta-analyses for re-analysis.

License GPL-2

Encoding UTF-8

Imports metafor, stats, stringr

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-04-10 09:05:21 UTC

R topics documented:

`format_CI` *Manuscript-friendly confidence interval formatting*

Description

Formats confidence interval lower and upper bounds into a rounded string.

Usage

```
format_CI(lo, hi, digits = 2)
```

Arguments

`lo`	Confidence interval lower limit (numeric)
`hi`	Confidence interval upper limit (numeric)
`digits`	Digits for rounding

Examples

```
format_CI(0.36, 0.72, 3)
```

`format_stat` *Manuscript-friendly number formatting*

Description

Formats a numeric result (e.g., p-value) as a manuscript-friendly string in which values below a minimum cutoff (e.g., 10^-5) are reported for example as "< 10^-5", values between the minimum cutoff and a maximum cutoff (e.g., 0.01) are reported in scientific notation, and p-values above the maximum cutoff are reported simply as, for example, 0.72.

Usage

```
format_stat(x, digits = 2, cutoffs = c(0.01, 10^-5))
```

Arguments

`x`	Numeric value to format
`digits`	Digits for rounding
`cutoffs`	A vector containing the two cutoffs

Examples

```
format_stat(0.735253)
format_stat(0.735253, digits = 4)
format_stat(0.0123)
format_stat(0.0001626)
format_stat(0.0001626, cutoffs = c(0.01, 10^-3))
```

---

parse_CI_string *Parse a string with point estimate and confidence interval*

Description

Given a vector of strings such as "0.65 (0.6, 0.70)", for example obtained by running optical character recognition (OCR) software on a screenshot of a published forest plot, parses the strings into a dataframe of point estimates and upper confidence interval limits. Assumes that the point estimate occurs before an opening bracket of the form "(" or "[" and that the confidence interval upper limit follows a the character `sep` (by default a comma, but might be a hyphen, for example). To further parse this dataframe into point estimates and variances, see `MetaUtility::scrape_meta`.

Usage

```
parse_CI_string(string, sep = ",")
```

Arguments

`string`	A vector of strings to be parsed.
`sep`	The character (not including whitespaces) separating the lower from the upper limits.

Examples

```
# messy string of confidence intervals
mystring = c( "0.65 [0.6, 0.7]", "0.8(0.5, 0.9]", "1.2  [0.3, 1.5)")
parse_CI_string(mystring)

# now with a hyphen separator
mystring = c( "0.65 [0.6- 0.7]", "0.8(0.5 - 0.9]", "1.2  [0.3 -1.5)")
parse_CI_string(mystring, sep="-")
```

`prop_stronger` *Estimate proportion of true effect sizes above or below a threshold*

Description

Estimates the proportion of true (i.e., population parameter) effect sizes in a meta-analysis that are above or below a specified threshold of scientific importance.

Usage

```
prop_stronger(q, M, t2, se.M = NA, se.t2 = NA, CI.level = 0.95,
  tail = NA, dat = NULL, R = 2000, bootstrap = "ifneeded",
  yi.name = "yi", vi.name = "vi")
```

Arguments

`q`	True effect size that is the threshold for "scientific importance"
`M`	Pooled point estimate from meta-analysis
`t2`	Estimated heterogeneity (tau^2) from meta-analysis
`se.M`	Estimated standard error of pooled point estimate from meta-analysis
`se.t2`	Estimated standard error 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`.
`dat`	Dataset of point estimates (with names equal to the passed `yi.name`) and their variances (with names equal to the passed `vi.name`). Only required when bootstrapping.
`R`	Number of bootstrap iterates. Only required when bootstrapping.
`bootstrap`	If equal to `ifneeded`, bootstraps if estimated proportion is less than 0.15 or more than 0.85. If equal to `never`, instead does not return inference in the above edge cases. Only required when bootstrapping.
`yi.name`	Name of the variable in `dat` containing the study-level point estimates. Only required when bootstrapping.
`vi.name`	Name of the variable in `dat` containing the study-level variances. Only required when bootstrapping.

Details

When the estimated proportion is less than 0.15 or more than 0.85, it is best to bootstrap the confidence interval using the bias-corrected and accelerated (BCa) method (Mathur & VanderWeele, 2018); this is the default behavior of `prop_stronger`. Sometimes BCa confidence interval estimation fails, in which case `prop_stronger` instead uses the percentile method, issuing a warning if this is the case. We use a modified "safe" version of the `boot` package code for bootstrapping such that if any bootstrap iterates fail (usually because of model estimation problems), the error message is printed but the bootstrap iterate is simply discarded so that confidence interval estimation can proceed.

Value

Returns a dataframe containing the point estimate for the proportion, its estimated standard error, and confidence interval limits.

References

1. Mathur MB & VanderWeele TJ. New metrics for meta-analyses of heterogeneous effects. Statistics in Medicine (2018).

2. Mathur MB & VanderWeele TJ. New metrics for multisite replication projects. Under review.

Examples

```
##### Example 1: BCG Vaccine and Tuberculosis Meta-Analysis #####

# calculate effect sizes for example dataset
d = metafor::escalc(measure="RR", ai=tpos, bi=tneg,
                   ci=cpos, di=cneg, data=metafor::dat.bcg)

# fit random-effects model
# note that metafor package returns on the log scale
m = metafor::rma.uni(yi= d$yi, vi=d$vi, knha=TRUE,
measure="RR", method="REML" )

# pooled point estimate (RR scale)
exp(m$b)

# estimate the proportion of effects stronger than RR = 0.80
# no bootstrapping will be needed
prop_stronger( q = log(0.8),
               M = as.numeric(m$b),
               t2 = m$tau2,
               se.M = as.numeric(m$vb),
               se.t2 = m$se.tau2,
               CI.level = 0.95,
              tail = "below",
              bootstrap = "ifneeded")

## Not run:
# now try a more extreme threshold, q, such that the function will use bootstrapping
# now we will need to pass the final 4 arguments as well
prop_stronger( q = log(0.9),
               M = as.numeric(m$b),
              t2 = m$tau2,
              se.M = as.numeric(m$vb),
              se.t2 = m$se.tau2,
              CI.level = 0.95,
              tail = "below",

              # below arguments control bootstrapping
              # only 100 iterates for demo purposes (should be higher in practice)
```

```
              dat = d,
              R = 100,
              bootstrap = "ifneeded",
              yi.name = "yi",
              vi.name = "vi" )

## End(Not run)

##### Example 2: Meta-Analysis of Multisite Replication Studies #####

 # replication estimates (Fisher's z scale) and SEs
 # from moral credential example in reference #2
 r.fis = c(0.303, 0.078, 0.113, -0.055, 0.056, 0.073,
 0.263, 0.056, 0.002, -0.106, 0.09, 0.024, 0.069, 0.074,
 0.107, 0.01, -0.089, -0.187, 0.265, 0.076, 0.082)

 r.SE = c(0.111, 0.092, 0.156, 0.106, 0.105, 0.057,
 0.091, 0.089, 0.081, 0.1, 0.093, 0.086, 0.076,
 0.094, 0.065, 0.087, 0.108, 0.114, 0.073, 0.105, 0.04)

 d = data.frame( yi = r.fis,
                 vi = r.SE^2 )

 # meta-analyze the replications
 m = metafor::rma.uni( yi = r.fis, vi = r.SE^2, measure = "ZCOR" )

 # probability of true effect above r = 0.10 = 28%
 # convert threshold on r scale to Fisher's z
 q = r_to_z(0.10)

 # bootstrap reps should be higher in practice (e.g., 1000)
 # here using only 100 for speed
 prop_stronger( q = q,
                M = m$b,
                se.M = m$se,
                t2 = m$tau2,
                se.t2 = m$se.tau2,
                tail = "above",
                dat = d,
                R = 250 )


 # probability of true effect equally strong in opposite direction
 q.star = r_to_z(-0.10)
 prop_stronger( q = q.star,
                M = m$b,
                se.M = m$se,
                t2 = m$tau2,
                se.t2 = m$se.tau2,
                tail = "below",
                dat = d,
                R = 250 )
```

round2 — *Round while keeping trailing zeroes*

Description

Rounds a numeric value and formats it as a string, keeping trailing zeroes.

Usage

```
round2(x, digits = 2)
```

Arguments

`x`	Numeric value to round
`digits`	Digits for rounding

Examples

```
round2(0.03000, digits = 4)

# compare to base round, which drops trailing zeroes and returns a numeric
round(0.03000, digits = 4)
```

r_to_d — *Convert Pearson's r to Cohen's d*

Description

Converts Pearson's r (computed with a continuous X and Y) to Cohen's d for use in meta-analysis. The resulting Cohen's d represents the estimated increase in standardized Y that is associated with a delta-unit increase in X.

Usage

```
r_to_d(r, sx, delta, N = NA, Ns = N, sx.known = FALSE)
```

Arguments

`r`	Pearson's correlation
`sx`	Sample standard deviation of X
`delta`	Contrast in X for which to compute Cohen's d, specified in raw units of X (not standard deviations).
`N`	Sample size used to estimate `r`
`Ns`	Sample size used to estimate `sx`, if different from `N`
`sx.known`	Is `sx` known rather than estimated? (By default, assumes `sx` is estimated, which will almost always be the case.)

Details

To preserve the sign of the effect size, the code takes the absolute value of `delta`. The standard error estimate assumes that X is approximately normal and that `N` is large.

References

1. Mathur MB & VanderWeele TJ. A simple, interpretable conversion from Pearson's correlation to Cohen's d for meta-analysis. Under revision.

Examples

```
# d for a 1-unit vs. a 2-unit increase in X
r_to_d( r = 0.5,
        sx = 2,
        delta = 1,
        N = 100 )
r_to_d( r = 0.5,
        sx = 2,
        delta = 2,
        N = 100 )

# d when sx is estimated in the same vs. a smaller sample
# point estimate will be the same, but inference will be a little
# less precise in second case
r_to_d( r = -0.3,
         sx = 2,
         delta = 2,
         N = 300,
         Ns = 300 )

r_to_d( r = -0.3,
        sx = 2,
        delta = 2,
        N = 300,
        Ns = 30 )
```

`r_to_z` *Convert Pearson's r to Fisher's z*

Description

Converts Pearson's r to Fisher's z for use in meta-analysis.

Usage

```
r_to_z(r)
```

Arguments

`r`	Pearson's correlation

Examples

```
# convert a Pearson correlation of -0.8 to Fisher's z
r_to_z(-0.8)
```

---

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

References

1. VanderWeele TJ (2017). On a square-root transformation of the odds ratio for a common outcome. Epidemiology.

tau_CI *Return confidence interval for tau for a meta-analysis*

Description

Returns confidence interval lower and upper limits for tau (the estimated standard deviation of the true effects) for a meta-analysis fit in `metafor::rma`.

Usage

```
tau_CI(meta, CI.level = 0.95)
```

Arguments

`meta`	A meta-analysis object fit in `metafor::rma`.
`CI.level`	Confidence interval level as a proportion (e.g., 0.95)

Examples

```
# calculate effect sizes for example dataset
d = metafor::escalc(measure="RR", ai=tpos, bi=tneg,
                   ci=cpos, di=cneg, data=metafor::dat.bcg)

# fit random-effects model
# note that metafor package returns on the log scale
m = metafor::rma.uni(yi= d$yi, vi=d$vi, knha=TRUE,
measure="RR", method="REML" )

tau_CI(m)

# for nicer formatting
format_CI( tau_CI(m)[1], tau_CI(m)[2] )
```

z_to_r *Convert Fisher's z to Pearson's r*

Description

Converts Fisher's z to Pearson's r for use in meta-analysis.

Usage

```
z_to_r(z)
```

Arguments

`z`	Fisher's z

Examples

```
# convert Fisher's z of 1.1 to Pearson's r
z_to_r(1.1)
```

Index