Package ‘Replicate’

October 12, 2022

Type Package

Title Statistical Metrics for Multisite Replication Studies

Version 1.2.0

Author Maya B. Mathur, Tyler J. VanderWeele

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

Description For a multisite replication project, computes the consistency metric P_orig, which is the probability that the original study would observe an estimated effect size as extreme or more extreme than it actually did, if in fact the original study were statistically consistent with the replications. Other recommended metrics are: (1) the probability of a true effect of scientifically meaningful size in the same direction as the estimate the original study; and (2) the probability of a true effect of meaningful size in the direction opposite the original study's estimate. These two can be computed using the package \code{MetaUtility::prop_stronger}. Additionally computes older metrics used in replication projects (namely expected agreement in ``statistical significance'' between an original study and replication studies as well as prediction intervals for the replication estimates). See Mathur and VanderWeele (under review; <https://osf.io/apnjk/>) for details.

License GPL-2

Imports metafor, stats, ggplot2

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-11-28 22:50:02 UTC

R topics documented:

`pred_int` *Compute prediction interval for replication study given original*

Description

Given point estimates and their variances for one or multiple original studies and one or more replication studies, returns a vector stating whether each replication estimate is in its corresponding prediction interval. Assumes no heterogeneity.

Usage

```
pred_int(yio, vio, yir = NULL, vir, level = 0.95)
```

Arguments

`yio`	Effect estimate in the original study. Can be a vector for multiple original studies.
`vio`	Estimated variance of effect estimate in the original study (i.e., its squared standard error). Can be a vector for multiple original studies.
`yir`	Effect estimate in the replication study. Can be a vector for multiple replication studies. Can be omitted, in which case function returns only the prediction interval.
`vir`	Estimated variance of effect estimate in the replication study (i.e., its squared standard error). Can be a vector for multiple replication studies.
`level`	Coverage level of prediction interval. Typically 0.95.

Examples

```
# calculate prediction interval for a single replication study
pred_int( yio = 1, vio = .5, yir = 0.6,
vir = .2 )

# calculate prediction intervals for a one-to-one design
pred_int( yio = c(1, 1.3), vio = c(.01, .6),
yir = c(.6, .7), vir = c(.01,.3) )

# no need to pass yir if you only want the intervals
pred_int( yio = c(1, 1.3), vio = c(.01, .6),
vir = c(.01,.3) )

# calculate prediction intervals for a many-to-one design
pred_int( yio = c(1), vio = c(.01), yir = c(.6, .7), vir = c(.01,.3) )
```

`prob_signif_agree` *Compute probability of "significance agreement" between replication and original study*

Description

Given point estimates and their variances for one or multiple original studies and variances for one or more replication studies, returns a vector of probabilities that the replication estimate is "statistically significant" and in the same direction as the original. Can be computed assuming no heterogeneity or allowing for heterogeneity.

Usage

```
prob_signif_agree(yio, vio, vir, t2 = 0, null = 0, alpha = 0.05)
```

Arguments

`yio`	Effect estimate in the original study. Can be a vector for multiple original studies.
`vio`	Estimated variance of effect estimate in the original study (i.e., its squared standard error). Can be a vector for multiple original studies.
`vir`	Estimated variance of effect estimate in the replication study (i.e., its squared standard error). Can be a vector for multiple replication studies.
`t2`	Optionally (if allowing for heterogeneity), the estimated variance of true effects across replication studies.
`null`	Null value for the hypothesis tests.
`alpha`	Alpha level for the hypothesis tests.

References

1. Mathur MB & VanderWeele TJ (under review). New statistical metrics for multisite replication projects.

Examples

```
# replication estimates (Fisher's z scale) and SEs
# from moral credential example in Mathur & VanderWeele
# (under review)
yir = 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)

seir = 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)

# how many do we expect to agree?
sum( prob_signif_agree( yio = 0.21, vio = 0.004, vir = seir^2 ) )
```

p_orig *Statistical consistency of original study with replication*

Description

Given the original study's effect estimate and its variance, the estimated average true effect size in the replications, and the estimated heterogeneity in the replications, computes estimated probability that the original study would have an effect estimate at least as extreme as the observed value if the original and the replications in fact are statistically consistent. Allows for heterogeneity.

Usage

```
p_orig(yio, vio, yr, t2, vyr)
```

Arguments

yio	Effect estimate in the original study.
vio	Estimated variance of effect estimate in the original study (i.e., its squared standard error).
yr	Estimated average true effect size in the replications.
t2	Estimated heterogeneity of true effect sizes in the replications.
vyr	Estimated variance of yr (i.e., its squared standard error).

Details

yr, vyr, and t2 can be estimated through, for example, random-effects meta-analysis or a mixed model fit to the individual subject data. See Mathur & VanderWeele's (under review) Appendix for details of how to specify such models.

References

1. Mathur MB & VanderWeele TJ (under review). New statistical metrics for multisite replication projects.

Examples

```
# replication estimates (Fisher's z scale) and SEs
# from moral credential example in Mathur and VanderWeele
# (under review)
yir = 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)

seir = 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)
```

```
# meta-analyze the replications
m = metafor::rma.uni( yi = yir, vi = seir^2, measure = "ZCOR" )

p_orig( yio = 0.210, vio = 0.062^2,
yr = m$b, t2 = m$se.tau2^2,  vyr = m$vb )
```

---

t2_lkl *Compute marginal log-likelihood of tau^2*

Description

Given point estimates and their variances, returns the marginal restricted log-likelihood of a specified tau^2 per Veroniki AA, et al. (2016), Section 3.11. Useful as a diagnostic for `p_orig` per Mathur VanderWeele (under review).

Usage

```
t2_lkl(yi, vi, t2)
```

Arguments

`yi`	Vector of point estimates
`vi`	Vector of variances of point estimates Can be a vector for multiple replication studies.
`t2`	Heterogeneity value (tau^2) at which to compute the marginal log-likelihood

References

1. Veroniki AA et al. (2016). Methods to estimate the between-study variance and its uncertainty in meta-analysis. *Research Synthesis Methods.*

2. Mathur MB & VanderWeele TJ (under review). New statistical metrics for multisite replication projects.

Examples

```
# replication estimates (Fisher's z scale) and SEs
# from moral credential example in Mathur & VanderWeele
# (under review)
yir = 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)

seir = 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)

vir = seir^2
```

```
# fit meta-analysis
.m = metafor::rma.uni( yi = yir,
              vi = vir,
              knha = TRUE )

# vector and list of tau^2 at which to compute the log-likelihood
t2.vec = seq(0, .m$tau2*10, .001)
t2l = as.list(t2.vec)

# compute the likelihood ratio vs. the MLE for each tau^2 in t2l
temp = lapply( t2l,
               FUN = function(t2) {
                 # log-lkl itself
                 t2_lkl( yi = yir,
                         vi = vir,
                         t2 = t2 )

                 # lkl ratio vs. the MLE
                 exp( t2_lkl( yi = yir,
                              vi = vir,
                              t2 = t2 ) ) / exp( t2_lkl( yi = yir,
                                                         vi = vir,
                                                         t2 = .m$tau2 ) )
               })

# plotting dataframe
dp = data.frame( tau = sqrt(t2.vec),
                 V = t2.vec,
                 lkl = unlist(temp) )

# fn: ratio of the plotted tau^2 vs. the actual MLE (for secondary x-axis)
g = function(x) x / .m$tau2

# breaks for main and secondary x-axes
breaks.x1 = seq( 0, max(dp$V), .005 )
breaks.x2 = seq( 0, max( g(dp$V) ), 1 )

p = ggplot2::ggplot( data = dp,
        ggplot2::aes(x = V,
            y = lkl ) ) +

  ggplot2::geom_vline(xintercept = .m$tau2,
             lty = 2,
             color = "red") +  # the actual MLE

  ggplot2::geom_line(lwd = 1.2) +
  ggplot2::theme_classic() +
  ggplot2::xlab( bquote( hat(tau)["*"]^2 ) ) +
 ggplot2::ylab( "Marginal likelihood ratio of " ~ hat(tau)["*"]^2 ~ " vs. " ~ hat(tau)^2 ) +
  ggplot2::scale_x_continuous( limits = c(0, max(breaks.x1)),
                      breaks = breaks.x1,
                      sec.axis = ggplot2::sec_axis( ~ g(.),
                                         name = bquote( hat(tau)["*"]^2 / hat(tau)^2 ),
```

```
                                                breaks=breaks.x2 ) ) +
      ggplot2::scale_y_continuous( limits = c(0,1),
                           breaks = seq(0,1,.1) )
    graphics::plot(p)
```

Index