Package 'PublicationBias'

January 20, 2023

Title Sensitivity Analysis for Publication Bias in Meta-Analyses

Version 2.3.0

Description Performs sensitivity analysis for publication bias in meta-analyses (per Mathur & VanderWeele, 2020 [<doi:10.31219/osf.io/s9dp6>]). These analyses enable statements such as: ``For publication bias to shift the observed point estimate to the null, 'significant' results would need to be at least 30-fold more likely to be published than negative or 'nonsignificant' results." Comparable statements can be made regarding shifting to a chosen non-null value or shifting the confidence interval. Provides a worst-case meta-analytic point estimate under maximal publication bias obtained simply by conducting a standard meta-analysis of only the negative and ``nonsignificant'' studies.

License GPL-2

URL https://github.com/mayamathur/PublicationBias, https://mayamathur.github.io/PublicationBias/

BugReports https://github.com/mayamathur/PublicationBias/issues

Encoding UTF-8

Depends R (>= 4.1.0)

Imports dplyr, ggplot2, glue, lifecycle, metabias, metafor, stats, Rdpack, rlang, robumeta

Suggests purrr, testthat (>= 3.0.0), withr

RoxygenNote 7.2.3

RdMacros Rdpack

Config/testthat/edition 3

NeedsCompilation no

Author Mika Braginsky [aut, cre], Maya Mathur [aut], Tyler J. VanderWeele [aut]

Maintainer Mika Braginsky <mika.br@gmail.com>

Repository CRAN

Date/Publication 2023-01-19 23:20:09 UTC

R topics documented:

`pubbias_meta` *Estimate publication bias-corrected meta-analysis*

Description

For a chosen ratio of publication probabilities, `selection_ratio`, estimates a publication bias-corrected pooled point estimate and confidence interval per Mathur and VanderWeele (2020). Model options include fixed-effects (a.k.a. "common-effect"), robust independent, and robust clustered specifications.

Usage

```
pubbias_meta(
  yi,
  vi,
  sei,
  cluster = 1:length(yi),
  selection_ratio,
  selection_tails = 1,
  model_type = "robust",
  favor_positive = TRUE,
  alpha_select = 0.05,
  ci_level = 0.95,
  small = TRUE,
  return_worst_meta = FALSE
)

corrected_meta(
  yi,
  vi,
  eta,
  clustervar = 1:length(yi),
  model,
  selection.tails = 1,
  favor.positive,
  alpha.select = 0.05,
  CI.level = 0.95,
  small = TRUE
)
```

Arguments

yi	A vector of point estimates to be meta-analyzed.
vi	A vector of estimated variances (i.e., squared standard errors) for the point estimates.
sei	A vector of estimated standard errors for the point estimates. (Only one of vi or sei needs to be specified).
cluster	Vector of the same length as the number of rows in the data, indicating which cluster each study should be considered part of (defaults to treating studies as independent; i.e., each study is in its own cluster).
selection_ratio	Ratio by which publication bias favors affirmative studies (i.e., studies with p-values less than alpha_select and estimates in the direction indicated by favor_positive).
selection_tails	1 (for one-tailed selection, recommended for its conservatism) or 2 (for two-tailed selection).
model_type	"fixed" for fixed-effects (a.k.a. "common-effect") or "robust" for robust random-effects.
favor_positive	TRUE if publication bias are assumed to favor significant positive estimates; FALSE if assumed to favor significant negative estimates.
alpha_select	Alpha level at which an estimate's probability of being favored by publication bias is assumed to change (i.e., the threshold at which study investigators, journal editors, etc., consider an estimate to be significant).
ci_level	Confidence interval level (as proportion) for the corrected point estimate. (The alpha level for inference on the corrected point estimate will be calculated from ci_level.)
small	Should inference allow for a small meta-analysis? We recommend always using TRUE.
return_worst_meta	Should the worst-case meta-analysis of only the nonaffirmative studies be returned?
eta	(deprecated) see selection_ratio
clustervar	(deprecated) see cluster
model	(deprecated) see model_type
selection.tails	(deprecated) see selection_tails
favor.positive	(deprecated) see favor_positive
alpha.select	(deprecated) see alpha_select
CI.level	(deprecated) see ci_level

Details

The `selection_ratio` represents the number of times more likely affirmative studies (i.e., those with a "statistically significant" and positive estimate) are to be published than nonaffirmative studies (i.e., those with a "nonsignificant" or negative estimate).

If `favor_positive` is `FALSE`, such that publication bias is assumed to favor negative rather than positive estimates, the signs of `yi` will be reversed prior to performing analyses. The corrected estimate will be reported based on the recoded signs rather than the original sign convention.

Value

An object of class `metabias::metabias()`, a list containing:

data A tibble with one row per study and the columns `yi`, `yif`, `vi`, `affirm`, `cluster`.

values A list with the elements `selection_ratio`, `selection_tails`, `model_type`, `favor_positive`, `alpha_select`, `ci_level`, `small`, `k`, `k_affirmative`, `k_nonaffirmative`.

stats A tibble with the columns `model`, `estimate`, `se`, `ci_lower`, `ci_upper`, `p_value`.

fit A list of fitted models, if any.

References

Mathur MB, VanderWeele TJ (2020). "Sensitivity analysis for publication bias in meta-analyses." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **69**(5), 1091–1119.

Examples

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

# first fit fixed-effects model without any bias correction
# since the point estimate is negative here, we'll assume publication bias
# favors negative log-RRs rather than positive ones
metafor::rma(yi, vi, data = dat, method = "FE")

# warmup
# note that passing selection_ratio = 1 (no publication bias) yields the naive
# point estimate from rma above, which makes sense
pubbias_meta(yi = dat$yi,
             vi = dat$vi,
             selection_ratio = 1,
             model_type = "fixed",
             favor_positive = FALSE)

# assume a known selection ratio of 5
# i.e., affirmative results are 5x more likely to be published than
# nonaffirmative ones
pubbias_meta(yi = dat$yi,
             vi = dat$vi,
             selection_ratio = 5,
```

```
             model_type = "fixed",
             favor_positive = FALSE)

# same selection ratio, but now account for heterogeneity and clustering via
# robust specification
pubbias_meta(yi = dat$yi,
             vi = dat$vi,
             cluster = dat$author,
             selection_ratio = 5,
             model_type = "robust",
             favor_positive = FALSE)

##### Make sensitivity plot as in Mathur & VanderWeele (2020) #####
# range of parameters to try (more dense at the very small ones)
selection_ratios <- c(200, 150, 100, 50, 40, 30, 20, seq(15, 1))

# compute estimate for each value of selection_ratio
estimates <- lapply(selection_ratios, function(e) {
  pubbias_meta(yi = dat$yi, vi = dat$vi, cluster = dat$author,
               selection_ratio = e, model_type = "robust",
               favor_positive = FALSE)$stats
})
estimates <- dplyr::bind_rows(estimates)
estimates$selection_ratio <- selection_ratios

require(ggplot2)
ggplot(estimates, aes(x = selection_ratio, y = estimate)) +
  geom_ribbon(aes(ymin = ci_lower, ymax = ci_upper), fill = "gray") +
  geom_line(lwd = 1.2) +
  labs(x = bquote(eta), y = bquote(hat(mu)[eta])) +
  theme_classic()
```

`pval_plot` *Plot one-tailed p-values*

Description

Plots the one-tailed p-values. The leftmost red line indicates the cutoff for one-tailed p-values less than 0.025 (corresponding to "affirmative" studies; i.e., those with a positive point estimate and a two-tailed p-value less than 0.05). The rightmost red line indicates one-tailed p-values greater than 0.975 (i.e., studies with a negative point estimate and a two-tailed p-value less than 0.05). If there is a substantial point mass of p-values to the right of the rightmost red line, this suggests that selection may be two-tailed rather than one-tailed.

Usage

```
pval_plot(yi, vi, sei, alpha_select = 0.05)
```

Arguments

yi	A vector of point estimates to be meta-analyzed. The signs of the estimates should be chosen such that publication bias is assumed to operate in favor of positive estimates.
vi	A vector of estimated variances (i.e., squared standard errors) for the point estimates.
sei	A vector of estimated standard errors for the point estimates. (Only one of vi or sei needs to be specified).
alpha_select	Alpha level at which an estimate's probability of being favored by publication bias is assumed to change (i.e., the threshold at which study investigators, journal editors, etc., consider an estimate to be significant).

References

Mathur MB, VanderWeele TJ (2020). "Sensitivity analysis for publication bias in meta-analyses." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **69**(5), 1091–1119.

Examples

```
# compute meta-analytic effect sizes
require(metafor)
dat <- metafor::escalc(measure = "RR", ai = tpos, bi = tneg, ci = cpos,
                       di = cneg, data = dat.bcg)

# flip signs since we think publication bias favors negative effects
dat$yi <- -dat$yi

pval_plot(yi = dat$yi, vi = dat$vi)
```

---

significance_funnel *Make significance funnel plot*

Description

Creates a modified funnel plot that distinguishes between affirmative and nonaffirmative studies, helping to detect the extent to which the nonaffirmative studies' point estimates are systematically smaller than the entire set of point estimates. The estimate among only nonaffirmative studies (gray diamond) represents a corrected estimate under worst-case publication bias. If the gray diamond represents a negligible effect size or if it is much smaller than the pooled estimate among all studies (black diamond), this suggests that the meta-analysis may not be robust to extreme publication bias. Numerical sensitivity analyses (via pubbias_svalue()) should still be carried out for more precise quantitative conclusions.

Usage

```
significance_funnel(
  yi,
  vi,
  sei,
  favor_positive = TRUE,
  alpha_select = 0.05,
  plot_pooled = TRUE,
  est_all = NA,
  est_worst = NA,
  xmin = min(yi),
  xmax = max(yi),
  ymin = 0,
  ymax = max(sqrt(vi)),
  xlab = "Point estimate",
  ylab = "Estimated standard error"
)
```

Arguments

yi	A vector of point estimates to be meta-analyzed.
vi	A vector of estimated variances (i.e., squared standard errors) for the point estimates.
sei	A vector of estimated standard errors for the point estimates. (Only one of vi or sei needs to be specified).
favor_positive	TRUE if publication bias are assumed to favor significant positive estimates; FALSE if assumed to favor significant negative estimates.
alpha_select	Alpha level at which an estimate's probability of being favored by publication bias is assumed to change (i.e., the threshold at which study investigators, journal editors, etc., consider an estimate to be significant).
plot_pooled	Should the pooled estimates within all studies and within only the nonaffirmative studies be plotted as well?
est_all	Regular meta-analytic estimate among all studies (optional).
est_worst	Worst-case meta-analytic estimate among only nonaffirmative studies (optional).
xmin	x-axis (point estimate) lower limit for plot.
xmax	x-axis (point estimate) upper limit for plot.
ymin	y-axis (standard error) lower limit for plot.
ymax	y-axis (standard error) upper limit for plot.
xlab	Label for x-axis (point estimate).
ylab	Label for y-axis (standard error).

Details

By default (`plot_pooled = TRUE`), also plots the pooled point estimate within all studies, supplied by the user as `est_all` (black diamond), and within only the nonaffirmative studies, supplied by the user as `est_worst` (gray diamond). The user can calculate `est_all` and `est_worst` using their choice of meta-analysis model. If instead these are not supplied but `plot_pooled = TRUE`, these pooled estimates will be automatically calculated using a fixed-effects (a.k.a. "common-effect") model.

References

Mathur MB, VanderWeele TJ (2020). "Sensitivity analysis for publication bias in meta-analyses." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **69**(5), 1091–1119.

Examples

```
##### Make Significance Funnel #####
# compute meta-analytic effect sizes for an example dataset
require(metafor)
dat <- metafor::escalc(measure = "RR", ai = tpos, bi = tneg, ci = cpos,
                       di = cneg, data = dat.bcg)

# favor_positive = FALSE since we think publication bias is in favor of negative
significance_funnel(yi = dat$yi, vi = dat$vi, favor_positive = FALSE)
```

Index