

Package: publipha (via r-universe)

October 26, 2024

Title Bayesian Meta-Analysis with Publications Bias and P-Hacking

Version 0.1.2

Description Tools for Bayesian estimation of meta-analysis models that account for publications bias or p-hacking. For publication bias, this package implements a variant of the p-value based selection model of Hedges (1992) <[doi:10.1214/ss/1177011364](https://doi.org/10.1214/ss/1177011364)> with discrete selection probabilities. It also implements the mixture of truncated normals model for p-hacking described in Moss and De Bin (2019) <[arXiv:1911.12445](https://arxiv.org/abs/1911.12445)>.

License GPL-3

Depends methods, R (>= 3.6.0), Rcpp (>= 0.12.19)

Imports rstan (>= 2.18.1), rstantools (>= 1.5.1), loo, truncnorm

LinkingTo BH (>= 1.72.0-2), Rcpp (>= 0.12.19), RcppEigen (>= 0.3.3.4.0), rstan (>= 2.21.8), StanHeaders (>= 2.21.0-7), RcppParallel

Encoding UTF-8

LazyData true

NeedsCompilation yes

SystemRequirements GNU make

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE, roclets = c("`rd", `` namespace", `` collate"))

Suggests testthat (>= 2.1.0), covr, knitr, rmarkdown, metafor, spelling, metadat

Collate 'data-anderson2010.R' 'data-baskerville2012.R' 'data-cuddy2018.R' 'data-dang2018.R' 'data-moty12017.R' 'densities-helpers.R' 'densities-mpsnorm.R' 'densities-phnorm.R' 'densities-psnorm.R' 'densities-snorm.R' 'ma.R' 'generics.R' 'publipha-package.R' 'stanmodels.R' 'tools.R' 'utility.R'

Language en-US

Repository <https://jonasmoss.r-universe.dev>

RemoteUrl <https://github.com/jonasmoss/publipha>

RemoteRef HEAD

RemoteSha 56099f20e2678d3cdb2b96a83cfd8b29dba4d842

Contents

publipha-package	2
dat.anderson2010	3
dat.baskerville2012	3
dat.cuddy2018	4
dat.dang2018	5
dat.motyl2017	6
ExtractParameters	6
loo,mafit-method	8
ma	8
mafit-class	11
mpsnorm	11
phnorm	13
psnorm	14
snorm	15
Index	17

publipha-package *The 'publipha' package.*

Description

Meta-analysis that corrects for publication selection bias and p-hacking.

References

Hedges, Larry V. "Modeling publication selection effects in meta-analysis." *Statistical Science* (1992): 246-255.

Moss, Jonas and De Bin, Riccardo. "Modelling publication bias and p-hacking" (2019) arXiv:1911.12445

Stan Development Team (2018). RStan: the R interface to Stan. R package version 2.18.1. <https://mc-stan.org>

dat.anderson2010 *Studies on Effect of Violent Video Games on Negative Outcomes*

Description

Results from 477 studies on the effect of violent video games on negative outcomes.

Usage

dat.anderson2010

Format

The data frame contains the following columns:

author	character	first author
year	numeric	publication year
outcome	character	one of seven outcomes
best	boolean	if TRUE, the was a best practice study
experimental	boolean	if TRUE, the study was experimental
adult	boolean	if TRUE, the study subjects were adults
country	character	country of study
ni	numeric	sample size
yi	numeric	observed mean difference in outcome (violent vs. non-violent)
vi	numeric	corresponding sampling variance

Source

<https://github.com/Joe-Hilgard/Anderson-meta>

References

Baskerville, N. B., Liddy, C., & Hogg, W. (2012). Systematic review and meta-analysis of practice facilitation. *Journal of General Internal Medicine*, 27(1), 1-10.

Hilgard, J., Engelhardt, C. R., & Rouder, J. N. (2017). Overstated evidence for short-term effects of violent video games on aggression. *Journal of Experimental Psychology: Applied*, 23(1), 1-10.

dat.baskerville2012 *Studies on Practice Facilitation*

Description

Results from 23 studies on the effect of practice facilitation in a primary care setting.

Usage

dat.baskerville2012

Format

The tibble contains the following columns:

author	character	first author of study
year	numeric	publication year
design	character	study design (RCT, C-RCT, or CCT)
blinded	boolean	if TRUE, the study was blinded
concealed	boolean	if TRUE, the study was concealed
yi	numeric	observed mean difference in outcome (facilitated vs non-facilitated)
vi	numeric	corresponding sampling variance

Source

Baskerville, N. B., Liddy, C., & Hogg, W. (2012). Systematic review and meta-analysis of practice facilitation within primary care settings. *The Annals of Family Medicine*, 10(1), 63-74.

dat.cuddy2018	<i>Studies on the Effect of Power Posing</i>
---------------	--

Description

Results from 27 studies related to power posing.

Usage

dat.cuddy2018

Format

The data frame contains the following columns:

author	character	first author
year	numeric	publication year
power	boolean	if TRUE, the outcome was feeling of power
ease	boolean	if TRUE, the outcome was an EASE variable
yi	numeric	standardized mean difference
vi	numeric	corresponding sampling variance

Details

The data points are taken from the p-curve analysis of Cuddy et al. (2018), restricted to 2 cell designs with mean difference as the outcome variable.

Source

<https://osf.io/jx3av/>

References

Cuddy, A. J., Schultz, S. J., & Fosse, N. E. (2018). P-curving a more comprehensive body of research on po

 dat.dang2018 *Meta-analysis on Ego Depletion*

Description

Results from 150 studies of ego depletion, the claim that self-control is a limited resource which is tapped whenever self-control is exerted.

Usage

```
data(dat.dang2018)
```

Format

The tibble contains the following columns:

author	character	the last name of the first author and the first letter of the last name of the second author;
year	numeric	publication year
in_carter	character	was the study in the meta-analysis of Carter et al. (2015)
study	character	the number given to the study in the original paper (0 = only one study was reported in the original)
dv	boolean	the dependent variable
iv	boolean	the independent variable
n1i	numeric	the number of participants in the depletion condition
n2i	numeric	the number of participants in the control condition
yi	numeric	the adjusted standardized mean difference
vi	numeric	the variance

Source

<https://link.springer.com/article/10.1007/s00426-017-0862-x#SupplementaryMaterial>

References

- Dang, J. (2018). An updated meta-analysis of the ego depletion effect. *Psychological Research*, 82(4), 645-651.
- Carter, E. C., Kofler, L. M., Forster, D. E., & McCullough, M. E. (2015). A series of meta-analytic tests of the depletion effect: self-control does not seem to rely on a limited resource. *Journal of Experimental Psychology: General*, 144(4), 796.

 dat.moty12017

Effect Sizes from 875 Studies in Psychology.

Description

Effect sizes from 875 studies in psychology. Adopted from Motyl et al. (2017).

Usage

```
data(dat.moty12017)
```

Format

The tibble contains the following columns:

author	character	first author of study
year	numeric	publication year
study	numeric	the number given to the study in the original paper (0 = only one study was reported in the orig
journal	character	journal where the study was published
concealed	character	design of the study; "Between", "Within", or "Mixed"
experimental	numeric	TRUE for an experimental study
ni	numeric	sample size
yi	numeric	observed mean difference in outcome
vi	numeric	corresponding sampling variance

Source

<https://osf.io/he8mu/>

References

Motyl, M., Demos, A. P., Carsel, T. S., Hanson, B. E., Melton, Z. J., Mueller, A. B., ... & Yantis, C. (2017). The state of social and personality science: Rotten to the core, not so bad, getting better, or getting worse?. *Journal of personality and social psychology*, 113(1), 34.

 ExtractParameters

Extract Parameters from an mafit Object

Description

Extract samples from a model of class `mafit` and apply a function `fun` to them.

Usage

```
extract_theta0(object, fun = mean)
extract_theta(object, fun = mean, i)
extract_tau(object, fun = mean)
extract_eta(object, fun = mean, i)
extract_isq(object, fun = mean)
```

Arguments

object	an object of class <code>mafit</code> .
fun	the function to be applied to the fitted parameters.
i	an optional index specifying which parameter to apply fun to. Only for <code>extract_eta</code> and <code>extract_theta</code> .

Details

Support parameters for extraction are: The meta-analytic mean θ_0 , the individual means θ , the heterogeneity parameter τ , the selection bias parameter η , and the I squared isq . See Higgins and Thompson (2002) for details about I squared.

All `extract_*` functions are wrappers around `rstan::extract`.

Value

The result of FUN being applied to all estimated parameters of object.

References

Higgins, J. P., & Thompson, S. G. (2002). Quantifying heterogeneity in a meta-analysis. *Statistics in medicine*, 21(11), 1539-1558.

Examples

```
set.seed(313)
model <- publipha::psma(yi = yi, vi = vi, data = dat.baskerville2012)
extract_theta0(model, mean) # [1] extract_theta0(model, mean)
extract_theta0(model, sd) # [1] 0.1095921
extract_tau(model, mean) # [1] 0.1315312
extract_theta(model, hist, i = 5)
```

loo,mafit-method	<i>Calculate the loo for an ma object.</i>
------------------	--

Description

Computes PSIS-LOO CV, approximate leave-one-out cross-validation using Pareto smoothed importance sampling, see [loo](#).

Usage

```
## S4 method for signature 'mafit'
loo(x, ...)
```

Arguments

x	an object of class <code>mafit</code> .
...	passed to loo . Only

Details

... affect the function through two parameters, `marginal` and `lower_bound`. When `marginalis` is TRUE, the PSIS-LOO CV is based on the marginal likelihood, i.e. with the dependence on θ integrated out. `marginal` defaults to TRUE. `lower_bound` specifies the lower bound where log-likelihoods are dropped; this is only used in the p -hacking model and defaults to -6.

Value

A [loo](#) object.

Examples

```
phma_model <- phma(yi, vi, data = metadat::dat.begg1989)
psma_model <- psma(yi, vi, data = metadat::dat.begg1989)
loo(phma_model)
loo(psma_model)
```

ma	<i>Meta-analysis Correcting for Publication Bias or p-hacking</i>
----	---

Description

Bayesian random effects meta-analysis. Correct for publication bias, correct for p -hacking, or run an ordinary meta-analysis without any correction.

Usage

```
ma(  
  yi,  
  vi,  
  bias = c("publication selection", "p-hacking", "none"),  
  data,  
  alpha = c(0, 0.025, 0.05, 1),  
  prior = NULL,  
  tau_prior = c("half-normal", "uniform", "inv_gamma"),  
  ...  
)  
  
psma(  
  yi,  
  vi,  
  data,  
  alpha = c(0, 0.025, 0.05, 1),  
  prior = NULL,  
  tau_prior = c("half-normal", "uniform", "inv_gamma"),  
  ...  
)  
  
phma(  
  yi,  
  vi,  
  data,  
  alpha = c(0, 0.025, 0.05, 1),  
  prior = NULL,  
  tau_prior = c("half-normal", "uniform", "inv_gamma"),  
  ...  
)  
  
cma(  
  yi,  
  vi,  
  data,  
  prior = NULL,  
  tau_prior = c("half-normal", "uniform", "inv_gamma"),  
  ...  
)  
  
allma(  
  yi,  
  vi,  
  data,  
  alpha = c(0, 0.025, 0.05, 1),  
  prior = NULL,  
  tau_prior = c("half-normal", "uniform", "inv_gamma"),
```

```
    ...
  )
```

Arguments

<code>yi</code>	Numeric vector of length <code>codek</code> with observed effect size estimates.
<code>vi</code>	Numeric vector of length <code>codek</code> with sampling variances.
<code>bias</code>	String; If "publication bias", corrects for publication bias. If "p-hacking", corrects for p-hacking.
<code>data</code>	Optional list or data frame containing <code>yi</code> and <code>vi</code> .
<code>alpha</code>	Numeric vector; Specifies the cutoffs for significance. Should include 0 and 1. Defaults to (0, 0.025, 0.05, 1).
<code>prior</code>	Optional list of prior parameters. See the details.
<code>tau_prior</code>	Which prior to use for <code>tau</code> , the heterogeneity parameter. Defaults to "half-normal"; "uniform" and "inv_gamma" are also supported.
<code>...</code>	Passed to <code>rstan::sampling</code> .

Details

`ma` does a Bayesian meta-analysis with the type of correction used specified by `bias`. `psma` is a wrapper for `ma` with `bias = "publication selection"`, `phma` is a wrapper with `bias = "p-hacking"`, while `cma` has `bias = "none"`. The function `allma` runs all bias options and returns a list.

The bias options are:

1. `publication selection`: The model of publication bias described in Hedges (1992).
2. `p-hacking`: The model for *p*-hacking described in Moss & De Bin (2019).
3. `none`: Classical random effects meta-analysis with no correction for selection bias.

The effect size distribution is normal with mean θ_0 and standard deviation τ . The prior for θ_0 is normal with parameters `theta0_mean` (default: 0), `theta0_sd` (default: 1). `eta` is the vector of *K* normalized publication probabilities (publication bias model) or *K* *p*-hacking probabilities (*p*-hacking model). The prior of `eta` is Dirichlet with parameter `eta0`, which defaults to `rep(1, K)` for the publication bias model and the *p*-hacking model. `eta0` is the prior for the Dirichlet distribution over the non-normalized `etas` in the publication bias model, and they are forced to be decreasing.

The standard prior for τ is half-normal with parameters `tau_mean` (default: 0), `tau_sd` (default: 1). If the uniform prior is used, the parameter are `u_min` (default: 0), and `u_max` with a default of 3. The inverse Gamma has parameters `shape` (default: 1) and `scale default`: 1.

To change the prior parameters, pass them to `prior` in a list.

Value

An S4 object of class `mafit` when `ma`, `psma`, `phma` or `cma` is run. A list of `mafit` objects when `allma` is run.

References

Hedges, Larry V. "Modeling publication selection effects in meta-analysis." *Statistical Science* (1992): 246-255.

Moss, Jonas and De Bin, Riccardo. "Modelling publication bias and p-hacking" (2019) arXiv:1911.12445

Examples

```
phma_model <- phma(yi, vi, data = metadat::dat.begg1989)

prior <- list(
  eta0 = c(3, 2, 1),
  theta0_mean = 0.5,
  theta0_sd = 10,
  tau_mean = 1,
  tau_sd = 1
)

psma_model <- psma(yi, vi, data = metadat::dat.begg1989, prior = prior)

cma_model <- psma(yi, vi, data = metadat::dat.begg1989, prior = prior)

model <- allma(yi, vi, data = metadat::dat.begg1989, prior = prior)
```

mafit-class

Class mafit: Fitted Meta-analysis Model

Description

Class mafit: Fitted Meta-analysis Model

mpsnorm

Marginal Publication Selection Meta-analysis Model

Description

Density, distribution, and random variate generation for the marginalized distribution of the publication selection meta-analysis model

Usage

```
dmpsnorm(x, theta0, tau, sigma, alpha = c(0, 0.025, 0.05, 1), eta, log = FALSE)
```

```
pmpsnorm(
  q,
  theta0,
  tau,
  sigma,
  alpha = c(0, 0.025, 0.05, 1),
  eta,
  lower.tail = TRUE,
  log.p = FALSE
)
```

```
rmpsnorm(n, theta0, tau, sigma, alpha = c(0, 0.025, 0.05, 1), eta)
```

Arguments

<code>x, q</code>	vector of quantiles.
<code>theta0</code>	vector of means.
<code>tau</code>	vector of heterogeneity parameters.
<code>sigma</code>	vector of study standard deviations.
<code>alpha</code>	vector of thresholds for publication bias.
<code>eta</code>	vector of publication probabilities, normalized to sum to 1.
<code>log, log.p</code>	logical; If TRUE, probabilities are given as $\log(p)$.
<code>lower.tail</code>	logical; If TRUE (default), the probabilities are $P[X \leq x]$ otherwise, $P[X \geq x]$.
<code>n</code>	number of observations. If $\text{length}(n) > 1$, the length is taken to be the number required.

Details

These functions assume a normal underlying effect size distribution and one-sided selection on the effects. For the fixed effects publication bias model see [psnorm](#).

Value

`dmpsnorm` gives the density, `pmpsnorm` gives the distribution function, and `rmpsnorm` generates random deviates.

References

Hedges, Larry V. "Modeling publication selection effects in meta-analysis." *Statistical Science* (1992): 246-255.

Moss, Jonas and De Bin, Riccardo. "Modelling publication bias and p-hacking" *Forthcoming* (2019)

Examples

```
rmpsnorm(100, theta0 = 0, tau = 0.1, sigma = 0.1, eta = c(1, 0.5, 0.1))
```

phnorm *p-hacking Meta-analysis Model*

Description

Density, distribution, and random variate generation for the p-hacking meta- analysis model.

Usage

```
dphnorm(x, theta, sigma, alpha = c(0, 0.025, 0.05, 1), eta, log = FALSE)
```

```
rphnorm(n, theta, sigma, alpha = c(0, 0.025, 0.05, 1), eta)
```

```
pphnorm(
  q,
  theta,
  sigma,
  alpha = c(0, 0.025, 0.05, 1),
  eta,
  lower.tail = TRUE,
  log.p = FALSE
)
```

Arguments

x, q	vector of quantiles.
theta	vector of means.
sigma	vector of study standard deviations.
alpha	vector of thresholds for p-hacking.
eta	vector of p-hacking probabilities, normalized to sum to 1.
log, log.p	logical; If TRUE, probabilities are given as log(p).
n	number of observations. If length(n) > 1, the length is taken to be the number required.
lower.tail	logical; If TRUE (default), the probabilities are $P[X \leq x]$ otherwise, $P[X \geq x]$.

Details

These functions assume one-sided selection on the effects. alpha contains the selection thresholds and eta the vector of p-hacking probabilities. theta is the true effect, while sigma is the true standard deviation before selection.

Value

dpsnorm gives the density, ppsnorm gives the distribution function, and rpsnorm generates random deviates.

References

Moss, Jonas and De Bin, Riccardo. "Modelling publication bias and p-hacking" Forthcoming (2019)

Examples

```
rpsnorm(100, theta = 0, sigma = 0.1, eta = c(1, 0.5, 0.1))
```

psnorm

Publication Selection Meta-analysis Model

Description

Density, distribution, quantile, random variate generation, and expectation calculation for the distribution for the publication selection meta-analysis model

Usage

```
dpsnorm(x, theta, sigma, alpha = c(0, 0.025, 0.05, 1), eta, log = FALSE)
```

```
ppsnorm(
  q,
  theta,
  sigma,
  alpha = c(0, 0.025, 0.05, 1),
  eta,
  lower.tail = TRUE,
  log.p = FALSE
)
```

```
rpsnorm(n, theta, sigma, alpha = c(0, 0.025, 0.05, 1), eta)
```

Arguments

x, q	vector of quantiles.
theta	vector of means.
sigma	vector of study standard deviations.
alpha	vector of thresholds for publication bias.
eta	vector of publication probabilities, normalized to sum to 1.
log, log.p	logical; If TRUE, probabilities are given as log(p).

`lower.tail` logical; If TRUE (default), the probabilities are $P[X \leq x]$ otherwise, $P[X \geq x]$.

`n` number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

The effect size distribution for the publication selection model is not normal, but has itself been selected for. These functions assume one-sided selection on the effects. These functions do not assume the existence of an underlying effect size distribution. For these, see [mpsnorm](#).

Value

`dpsnorm` gives the density, `ppsnorm` gives the distribution function, and `rpsnorm` generates random deviates.

References

Hedges, Larry V. "Modeling publication selection effects in meta-analysis." *Statistical Science* (1992): 246-255.

Moss, Jonas and De Bin, Riccardo. "Modelling publication bias and p-hacking" *Forthcoming* (2019)

Examples

```
rpsnorm(100, theta = 0, sigma = 0.1, eta = c(1, 0.5, 0.1))
```

snorm *Selected Normal Effect Size Distribution*

Description

Density, random variate generation, and expectation calculation for the effect size distribution of the one-sided normal publication bias model.

Usage

```
dsnrm(x, theta0, tau, sigma, alpha = c(0, 0.025, 0.05, 1), eta, log = FALSE)
```

```
rsnorm(n, theta0, tau, sigma, alpha = c(0, 0.025, 0.05, 1), eta)
```

```
esnorm(theta0, tau, sigma, alpha, eta)
```

Arguments

x	vector of quantiles.
theta0	vector of means.
tau	vector of heterogeneity parameters.
sigma	vector of study standard deviations.
alpha	vector of thresholds for publication bias.
eta	vector of publication probabilities, normalized to sum to 1.
log	logical; If TRUE, probabilities are given as $\log(p)$.
n	number of observations. If $\text{length}(n) > 1$, the length is taken to be the number required.

Details

The effect size distribution for the publication selection model is not normal, but has itself been selected for. These functions assume a normal underlying effect size distribution and one-sided selection on the effects.

Value

dsnrm gives the density, psnorm gives the distribution function, and rsnrm generates random deviates.

References

Hedges, Larry V. "Modeling publication selection effects in meta-analysis." *Statistical Science* (1992): 246-255.

Moss, Jonas and De Bin, Riccardo. "Modelling publication bias and p-hacking" (2019) arXiv:1911.12445

Examples

```
rsnorm(100, theta0 = 0, tau = 0.1, sigma = 0.1, eta = c(1, 0.5, 0.1))
```


Index

* datasets

- dat.anderson2010, 3
- dat.baskerville2012, 3
- dat.cuddy2018, 4
- dat.dang2018, 5
- dat.motyl2017, 6

allma (ma), 8

cma (ma), 8

- dat.anderson2010, 3
- dat.baskerville2012, 3
- dat.cuddy2018, 4
- dat.dang2018, 5
- dat.motyl2017, 6
- dmpsnorm (mpsnorm), 11
- dphnorm (phnorm), 13
- dpsnorm (psnorm), 14
- dsnorm (snorm), 15

- esnorm (snorm), 15
- extract_eta (ExtractParameters), 6
- extract_isq (ExtractParameters), 6
- extract_tau (ExtractParameters), 6
- extract_theta (ExtractParameters), 6
- extract_theta0 (ExtractParameters), 6
- ExtractParameters, 6

loo, 8

loo,mafit-method, 8

ma, 8

mafit, 6, 7

mafit-class, 11

mpsnorm, 11, 15

phma (ma), 8

phnorm, 13

pmpsnorm (mpsnorm), 11

pphnorm (phnorm), 13

ppsnorm (psnorm), 14

psma (ma), 8

psnorm, 12, 14

publipha (publipha-package), 2

publipha-package, 2

rmpsnorm (mpsnorm), 11

rphnorm (phnorm), 13

rpsnorm (psnorm), 14

rsnorm (snorm), 15

rstan::extract, 7

snorm, 15