

Package ‘metaBMA’

June 2, 2020

Type Package

Date 2020-05-30

Title Bayesian Model Averaging for Random and Fixed Effects
Meta-Analysis

Version 0.6.3

Description Computes the posterior model probabilities for standard meta-analysis models (null model vs. alternative model assuming either fixed- or random-effects, respectively). These posterior probabilities are used to estimate the overall mean effect size as the weighted average of the mean effect size estimates of the random- and fixed-effect model as proposed by Gronau, Van Erp, Heck, Cesario, Jonas, & Wagenmakers (2017, <doi:10.1080/23743603.2017.1326760>). The user can define a wide range of non-informative or informative priors for the mean effect size and the heterogeneity coefficient. Moreover, using pre-compiled Stan models, meta-analysis with continuous and discrete moderators with Jeffreys-Zellner-Siow (JZS) priors can be fitted and tested. This allows to compute Bayes factors and perform Bayesian model averaging across random- and fixed-effects meta-analysis with and without moderators. For a primer on Bayesian model-averaged meta-analysis, see Gronau, Heck, Berkhout, Haaf, & Wagenmakers (2020, <doi:10.31234/osf.io/97qup>).

Depends R (>= 3.4.0), Rcpp (>= 1.0.0), methods

Imports mvtnorm, logspline, coda, LaplacesDemon, rstan (>= 2.18.1),
rstantools (>= 1.5.1), bridgesampling

Suggests testthat, knitr

LinkingTo BH (>= 1.69.0-1), Rcpp (>= 1.0.0), RcppEigen (>= 0.3.3.5.0),
rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

VignetteBuilder knitr

URL <https://github.com/danheck/metaBMA>

License GPL-3

Encoding UTF-8

LazyData true

NeedsCompilation yes

SystemRequirements GNU make

RoxygenNote 7.1.0

Author Daniel W. Heck [aut, cre] (<<https://orcid.org/0000-0002-6302-9252>>),
 Quentin F. Gronau [ctb],
 Eric-Jan Wagenmakers [ctb]

Maintainer Daniel W. Heck <dheck@uni-marburg.de>

Repository CRAN

Date/Publication 2020-06-02 08:40:02 UTC

R topics documented:

metaBMA-package	2
bma	4
facial_feedback	5
inclusion	6
meta_bma	7
meta_default	9
meta_fixed	11
meta_ordered	12
meta_random	15
plot.meta_pred	17
plot.prior	17
plot_default	18
plot_forest	19
plot_posterior	20
power_pose	20
predicted_bf	22
prior	22
towels	24

Index	25
--------------	-----------

metaBMA-package	<i>metaBMA: Bayesian Model Averaging for Random and Fixed Effects Meta-Analysis</i>
-----------------	---

Description

Fixed-effects meta-analyses assume that the effect size d is identical in all studies. In contrast, random-effects meta-analyses assume that effects vary according to a normal distribution with mean d and standard deviation τ . Both models can be compared in a Bayesian framework by assuming specific prior distribution for d and τ (see [prior](#)). Given the posterior model probabilities, the evidence for or against an effect (i.e., whether $d = 0$) and the evidence for or against random effects can be evaluated (i.e., whether $\tau = 0$). By using Bayesian model averaging, both tests can be performed by integrating over the other model. This allows to test whether an effect exists while accounting for uncertainty whether study heterogeneity exists (so-called inclusion Bayes factors). For a primer on Bayesian model-averaged meta-analysis, see Gronau, Heck, Berkhout, Haaf, and Wagenmakers (2020).

Details

The most general functions in metaBMA is `meta_bma`, which fits random- and fixed-effects models, compute the inclusion Bayes factor for the presence of an effect and the averaged posterior distribution of the mean effect d (which accounts for uncertainty regarding study heterogeneity). Prior distributions can be specified and plotted using the function `prior`.

Moreover, `meta_fixed` and `meta_random` fit a single meta-analysis models. The model-specific posteriors for d can be averaged by `bma` and inclusion Bayes factors be computed by `inclusion`.

Results can be visualized with the functions `plot_posterior`, which compares the prior and posterior density for a fitted meta-analysis, and `plot_forest`, which plots study and overall effect sizes.

For more details how to use the package, see the vignette: `vignette("metaBMA")`.

Funding

Funding for this research was provided by the Berkeley Initiative for Transparency in the Social Sciences, a program of the Center for Effective Global Action (CEGA), Laura and John Arnold Foundation, and by the German Research Foundation (grant GRK-2277: Statistical Modeling in Psychology).

Author(s)

Heck, D. W. & Gronau, Q. F.

References

Gronau, Q. F., Erp, S. V., Heck, D. W., Cesario, J., Jonas, K. J., & Wagenmakers, E.-J. (2017). A Bayesian model-averaged meta-analysis of the power pose effect with informed and default priors: the case of felt power. *Comprehensive Results in Social Psychology*, 2(1), 123-138. <https://doi.org/10.1080/23743603.2017.1326760>

Gronau, Q. F., Heck, D. W., Berkhout, S. W., Haaf, J. M., & Wagenmakers, E.-J. (2020). A primer on Bayesian model-averaged meta-analysis. <https://doi.org/10.31234/osf.io/97qup>

Heck, D. W., Gronau, Q. F., & Wagenmakers, E.-J. (2018). metaBMA: Bayesian model averaging for random and fixed effects meta-analysis. R package version 0.5.0. <https://github.com/danheck/metaBMA>. doi:10.5281/zenodo.835494

See Also

Useful links:

- <https://github.com/danheck/metaBMA>

Description

Model averaging for different meta-analysis models (e.g., random-effects or fixed-effects with different priors) based on the posterior model probability.

Usage

```
bma(
  meta,
  prior = 1,
  parameter = "d",
  summarize = "integrate",
  ci = 0.95,
  rel.tol = .Machine$double.eps^0.5
)
```

Arguments

<code>meta</code>	list of meta-analysis models (fitted via meta_random or meta_fixed)
<code>prior</code>	prior probabilities over models (possibly unnormalized). For instance, if the first model is as likely as models 2, 3 and 4 together: <code>prior = c(3, 1, 1, 1)</code> . The default is a discrete uniform distribution over models.
<code>parameter</code>	either the mean effect "d" or the heterogeneity "tau" (i.e., the across-study standard deviation of population effect sizes).
<code>summarize</code>	how to estimate parameter summaries (mean, median, SD, etc.): Either by numerical integration (<code>summarize = "integrate"</code>) or based on MCMC/Stan samples (<code>summarize = "stan"</code>).
<code>ci</code>	probability for the credibility/highest-density intervals.
<code>rel.tol</code>	relative tolerance used for numerical integration using integrate . Use <code>rel.tol=.Machine\$double.eps</code> for maximal precision (however, this might be slow).

Examples

```
# model averaging for fixed and random effects
data(towels)
fixed <- meta_fixed(logOR, SE, study, towels)
random <- meta_random(logOR, SE, study, towels, iter = 1000)

averaged <- bma(list("fixed" = fixed, "random" = random))
averaged
plot_posterior(averaged)
plot_forest(averaged, mar = c(4.5, 20, 4, .3))
```

`facial_feedback`*Data Set: Facial Feedback*

Description

Preregistered replication (Wagenmakers et al., 2016) that investigated the facial feedback hypothesis (Strack, Martin, & Stepper, 1988).

Usage

```
facial_feedback
```

Format

A data frame with three variables:

`study` Authors of original study (see Wagenmakers et. al, 2016)

`d` Measure of effect size: Cohen's d (difference between smile vs. pout condition)

`SE` Measure of precision: standard error of Cohen's d

Details

The facial-feedback hypothesis states that people's affective responses can be influenced by their own facial expression (e.g., smiling, pouting), even when their expression did not result from their emotional experiences (Strack, Martin, & Stepper, 1988).

References

Strack, F., Martin, L. L., & Stepper, S. (1988). Inhibiting and facilitating conditions of the human smile: A nonobtrusive test of the facial feedback hypothesis. *Journal of Personality and Social Psychology*, 54, 768–777. <https://doi.org/10.1037/0022-3514.54.5.768>

Wagenmakers, E.-J., Beek, T., Dijkhoff, L., Gronau, Q. F., Acosta, A., Adams, R. B., ... Zwaan, R. A. (2016). Registered replication report: Strack, Martin, & Stepper (1988). *Perspectives on Psychological Science*, 11, 917-928. <https://doi.org/10.1177/1745691616674458>

Examples

```
data(facial_feedback)
head(facial_feedback)
mf <- meta_fixed(d, SE, study, facial_feedback)
mf
plot_posterior(mf)
```

inclusion	<i>Inclusion Bayes Factor</i>
-----------	-------------------------------

Description

Computes the inclusion Bayes factor for two sets of models (e.g., $A=\{M1,M2\}$ vs. $B=\{M3,M4\}$).

Usage

```
inclusion(logml, include = 1, prior = 1)
```

Arguments

logml	a vector with log-marginal likelihoods. Alternatively, a list with meta-analysis models (fitted via meta_random or meta_fixed).
include	integer vector which models to include in inclusion Bayes factor/posterior probability. If only two marginal likelihoods/meta-analyses are supplied, the inclusion Bayes factor is identical to the usual Bayes factor $BF_{\{M1,M2\}}$. One can include models depending on the names of the models (such as "random_H1") by providing a character value, for instance: <code>include="H1"</code> (all H1 vs. all H0 models) or <code>include="random"</code> (all random- vs. all fixed-effects models).
prior	prior probabilities over models (possibly unnormalized). For instance, if the first model is as likely as models 2, 3 and 4 together: <code>prior = c(3, 1, 1, 1)</code> . The default is a discrete uniform distribution over models.

Examples

```
#### Example with simple Normal-distribution models
# generate data:
x <- rnorm(50)

# Model 1: x ~ Normal(0,1)
logm1 <- sum(dnorm(x, log = TRUE))
# Model 2: x ~ Normal(.2, 1)
logm2 <- sum(dnorm(x, mean = .2, log = TRUE))
# Model 3: x ~ Student-t(df=2)
logm3 <- sum(dt(x, df=2, log = TRUE))

# BF: Correct (Model 1) vs. misspecified (2 & 3)
inclusion(c(logm1, logm2, logm3), include = 1)
```

Description

Fits random- and fixed-effects meta-analyses and performs Bayesian model averaging for H1 ($d \neq 0$) vs. H0 ($d = 0$).

Usage

```
meta_bma(
  y,
  SE,
  labels,
  data,
  d = prior("norm", c(mean = 0, sd = 0.3)),
  tau = prior("invgamma", c(shape = 1, scale = 0.15)),
  rscale_contin = 1/2,
  rscale_discrete = sqrt(2)/2,
  centering = TRUE,
  prior = c(1, 1, 1, 1),
  logml = "integrate",
  summarize = "stan",
  ci = 0.95,
  rel.tol = .Machine$double.eps^0.3,
  logml_iter = 5000,
  silent_stan = TRUE,
  ...
)
```

Arguments

y	effect size per study. Can be provided as (1) a numeric vector, (2) the quoted or unquoted name of the variable in data, or (3) a formula to include discrete or continuous moderator variables.
SE	standard error of effect size for each study. Can be a numeric vector or the quoted or unquoted name of the variable in data
labels	optional: character values with study labels. Can be a character vector or the quoted or unquoted name of the variable in data
data	data frame containing the variables for effect size y, standard error SE, labels, and moderators per study.
d	prior distribution on the average effect size d. The prior probability density function is defined via prior .
tau	prior distribution on the between-study heterogeneity tau (i.e., the standard deviation of the study effect sizes d_{study} in a random-effects meta-analysis. A (nonnegative) prior probability density function is defined via prior .

rscale_contin	scale parameter of the JZS prior for the continuous covariates.
rscale_discrete	scale parameter of the JZS prior for discrete moderators.
centering	whether continuous moderators are centered.
prior	prior probabilities over models (possibly unnormalized) in the order <code>c(fixed_H0, fixed_H1, random_H0, ...)</code> . For instance, if we expect fixed effects to be two times as likely as random effects and H0 and H1 to be equally likely: <code>prior = c(2, 2, 1, 1)</code> .
logml	how to estimate the log-marginal likelihood: either by numerical integration ("integrate") or by bridge sampling using MCMC/Stan samples ("stan"). To obtain high precision with <code>logml="stan"</code> , many MCMC samples are required (e.g., <code>logml_iter=10000, warmup=1000</code>).
summarize	how to estimate parameter summaries (mean, median, SD, etc.): Either by numerical integration (<code>summarize = "integrate"</code>) or based on MCMC/Stan samples (<code>summarize = "stan"</code>).
ci	probability for the credibility/highest-density intervals.
rel.tol	relative tolerance used for numerical integration using <code>integrate</code> . Use <code>rel.tol=.Machine\$double.eps</code> for maximal precision (however, this might be slow).
logml_iter	number of iterations (per chain) from the posterior distribution of d and τ . The samples are used for computing the marginal likelihood of the random-effects model with bridge sampling (if <code>logml="stan"</code>) and for obtaining parameter estimates (if <code>summarize="stan"</code>). Note that the argument <code>iter=2000</code> controls the number of iterations for estimation of the random-effect parameters per study in random-effects meta-analysis.
silent_stan	whether to suppress the Stan progress bar.
...	further arguments passed to <code>rstan::sampling</code> (see stanmodel-method-sampling). Relevant MCMC settings concern the number of warmup samples that are discarded (<code>warmup=500</code>), the total number of iterations per chain (<code>iter=2000</code>), the number of MCMC chains (<code>chains=4</code>), whether multiple cores should be used (<code>cores=4</code>), and control arguments that make the sampling in Stan more robust, for instance: <code>control=list(adapt_delta=.97)</code> .

Details

Bayesian model averaging for four meta-analysis models: Fixed- vs. random-effects and H0 ($d = 0$) vs. H1 (e.g., $d > 0$). For a primer on Bayesian model-averaged meta-analysis, see Gronau, Heck, Berkhout, Haaf, and Wagenmakers (2020).

By default, the log-marginal likelihood is computed by numerical integration (`logml="integrate"`). This is relatively fast and gives precise, reproducible results. However, for extreme priors or data (e.g., very small standard errors), numerical integration is not robust and might provide incorrect results. As an alternative, the log-marginal likelihood can be estimated using MCMC/Stan samples and bridge sampling (`logml="stan"`).

To obtain posterior summary statistics for the average effect size d and the heterogeneity parameter τ , one can also choose between numerical integration (`summarize="integrate"`) or MCMC sampling in Stan (`summarize="stan"`). If any moderators are included in a model, both the marginal likelihood and posterior summary statistics can only be computed using Stan.

References

Gronau, Q. F., Erp, S. V., Heck, D. W., Cesario, J., Jonas, K. J., & Wagenmakers, E.-J. (2017). A Bayesian model-averaged meta-analysis of the power pose effect with informed and default priors: the case of felt power. *Comprehensive Results in Social Psychology*, 2(1), 123-138. <https://doi.org/10.1080/23743603.2017.1326760>

Gronau, Q. F., Heck, D. W., Berkhout, S. W., Haaf, J. M., & Wagenmakers, E.-J. (2020). A primer on Bayesian model-averaged meta-analysis. <https://doi.org/10.31234/osf.io/97qup>

See Also

[meta_fixed](#), [meta_random](#)

Examples

```
# Note: The following example optimizes speed (for CRAN checks).
#       The settings are not suitable for actual data analysis!

data(towels)
set.seed(123)
mb <- meta_bma(logOR, SE, study, towels,
               d = prior("norm", c(mean=0, sd=.3), lower=0),
               tau = prior("invgamma", c(shape = 1, scale = 0.15)),
               rel.tol = .Machine$double.eps^.15, iter=1000)

mb
plot_posterior(mb, "d")
```

meta_default

Defaults for Model Averaging in Meta-Analysis

Description

Wrapper with default prior for Bayesian meta-analysis based on a literature review. Currently, the same default is used in all cases.

Usage

```
meta_default(y, SE, labels, data, field = "psychology", effect = "d", ...)
```

Arguments

y	effect size per study. Can be provided as (1) a numeric vector, (2) the quoted or unquoted name of the variable in data, or (3) a formula to include discrete or continuous moderator variables.
SE	standard error of effect size for each study. Can be a numeric vector or the quoted or unquoted name of the variable in data
labels	optional: character values with study labels. Can be a character vector or the quoted or unquoted name of the variable in data

data	data frame containing the variables for effect size y , standard error SE, labels, and moderators per study.
field	either "psychology" or "medicine" (uses partial matching, so "p" and "m" are sufficient)
effect	the type of effect size: either Cohen's d ("d"), Pearson correlations ("r"), Fisher's z -transformed correlations ("z"), or log-odds ratios ("logOR").
...	further arguments passed to <code>meta_bma</code>

Details

Default prior distributions can be plotted using `plot_default`.

For `field = "psychology"`, the following defaults are used:

- `effect = "d"`: Half-normal with $SD=0.3$ on mean effect and half-Cauchy with $scale=.5$ on standard deviation of effects.
- `effect = "r"`: Half-normal with $SD=0.3$ on mean effect and half-Cauchy with $scale=.5$ on standard deviation of effects.
- `effect = "z"`: Half-normal with $SD=0.3$ on mean effect and half-Cauchy with $scale=.5$ on standard deviation of effects.
- `effect = "logOR"`: Half-normal with $SD=0.3$ on mean effect and half-Cauchy with $scale=.5$ on standard deviation of effects.

For `field = "medicine"`, the following defaults are used:

- `effect = "d"`: Half-normal with $SD=0.3$ on mean effect and half-Cauchy with $scale=.5$ on standard deviation of effects.
- `effect = "r"`: Half-normal with $SD=0.3$ on mean effect and half-Cauchy with $scale=.5$ on standard deviation of effects.
- `effect = "z"`: Half-normal with $SD=0.3$ on mean effect and half-Cauchy with $scale=.5$ on standard deviation of effects.
- `effect = "logOR"`: Half-normal with $SD=0.3$ on mean effect and half-Cauchy with $scale=.5$ on standard deviation of effects.

References

Gronau, Q. F., Erp, S. V., Heck, D. W., Cesario, J., Jonas, K. J., & Wagenmakers, E.-J. (2017). A Bayesian model-averaged meta-analysis of the power pose effect with informed and default priors: the case of felt power. *Comprehensive Results in Social Psychology*, 2(1), 123-138. <https://doi.org/10.1080/23743603.2017.1326760>

See Also

`meta_bma`, `plot_default`

Examples

```
# Note: The following example optimizes speed (for CRAN checks).
#       The settings are not suitable for actual data analysis!

data(towels)
set.seed(123)
md <- meta_default(logOR, SE, study, towels,
                  field = "psych", effect = "logOR",
                  rel.tol=.Machine$double.eps^.15, iter=1000)

md
plot_forest(md)
```

meta_fixed

*Bayesian Fixed-Effects Meta-Analysis***Description**

Runs a Bayesian meta-analysis assuming that the mean effect d in each study is identical (i.e., a fixed-effects analysis).

Usage

```
meta_fixed(
  y,
  SE,
  labels,
  data,
  d = prior("norm", c(mean = 0, sd = 0.3)),
  rscale_contin = 1/2,
  rscale_discrete = sqrt(2)/2,
  centering = TRUE,
  logml = "integrate",
  summarize = "integrate",
  ci = 0.95,
  rel.tol = .Machine$double.eps^0.3,
  silent_stan = TRUE,
  ...
)
```

Arguments

y effect size per study. Can be provided as (1) a numeric vector, (2) the quoted or unquoted name of the variable in `data`, or (3) a [formula](#) to include discrete or continuous moderator variables.

SE standard error of effect size for each study. Can be a numeric vector or the quoted or unquoted name of the variable in `data`

labels	optional: character values with study labels. Can be a character vector or the quoted or unquoted name of the variable in data
data	data frame containing the variables for effect size y , standard error SE, labels, and moderators per study.
d	prior distribution on the average effect size d . The prior probability density function is defined via prior .
rscale_contin	scale parameter of the JZS prior for the continuous covariates.
rscale_discrete	scale parameter of the JZS prior for discrete moderators.
centering	whether continuous moderators are centered.
logml	how to estimate the log-marginal likelihood: either by numerical integration ("integrate") or by bridge sampling using MCMC/Stan samples ("stan"). To obtain high precision with logml="stan", many MCMC samples are required (e.g., logml_iter=10000, warmup=1000).
summarize	how to estimate parameter summaries (mean, median, SD, etc.): Either by numerical integration (summarize = "integrate") or based on MCMC/Stan samples (summarize = "stan").
ci	probability for the credibility/highest-density intervals.
rel.tol	relative tolerance used for numerical integration using integrate . Use rel.tol=.Machine\$double.eps for maximal precision (however, this might be slow).
silent_stan	whether to suppress the Stan progress bar.
...	further arguments passed to <code>rstan::sampling</code> (see stanmodel-method-sampling). Relevant MCMC settings concern the number of warmup samples that are discarded (warmup=500), the total number of iterations per chain (iter=2000), the number of MCMC chains (chains=4), whether multiple cores should be used (cores=4), and control arguments that make the sampling in Stan more robust, for instance: <code>control=list(adapt_delta=.97)</code> .

Examples

```
data(towels)
### Bayesian Fixed-Effects Meta-Analysis (H1: d>0 Cauchy)
mf <- meta_fixed(logOR, SE, study, data = towels,
                 d = prior("norm", c(mean=0, sd=.3), lower=0))
mf
plot_posterior(mf)
plot_forest(mf)
```

meta_ordered

Meta-Analysis with Order-Constrained Study Effects

Description

Computes the Bayes factor for the hypothesis that the true study effects in a random-effects meta-analysis are all positive or negative.

Usage

```
meta_ordered(
  y,
  SE,
  labels,
  data,
  d = prior("norm", c(mean = 0, sd = 0.3), lower = 0),
  tau = prior("invgamma", c(shape = 1, scale = 0.15)),
  prior = c(1, 1, 1, 1),
  logml = "integrate",
  summarize = "stan",
  ci = 0.95,
  rel.tol = .Machine$double.eps^0.3,
  logml_iter = 5000,
  iter = 5000,
  silent_stan = TRUE,
  ...
)
```

Arguments

y	effect size per study. Can be provided as (1) a numeric vector, (2) the quoted or unquoted name of the variable in data, or (3) a formula to include discrete or continuous moderator variables.
SE	standard error of effect size for each study. Can be a numeric vector or the quoted or unquoted name of the variable in data
labels	optional: character values with study labels. Can be a character vector or the quoted or unquoted name of the variable in data
data	data frame containing the variables for effect size y, standard error SE, labels, and moderators per study.
d	prior distribution on the average effect size d. The prior probability density function is defined via prior .
tau	prior distribution on the between-study heterogeneity tau (i.e., the standard deviation of the study effect sizes d_{study} in a random-effects meta-analysis. A (nonnegative) prior probability density function is defined via prior .
prior	prior probabilities over models (possibly unnormalized) in the order <code>c(fixed_H0, fixed_H1, ordered_H1)</code> . Note that the model <code>random_H0</code> is not included in the comparison.
logml	how to estimate the log-marginal likelihood: either by numerical integration ("integrate") or by bridge sampling using MCMC/Stan samples ("stan"). To obtain high precision with <code>logml="stan"</code> , many MCMC samples are required (e.g., <code>logml_iter=10000, warmup=1000</code>).
summarize	how to estimate parameter summaries (mean, median, SD, etc.): Either by numerical integration (<code>summarize = "integrate"</code>) or based on MCMC/Stan samples (<code>summarize = "stan"</code>).
ci	probability for the credibility/highest-density intervals.

rel.tol	relative tolerance used for numerical integration using integrate . Use <code>rel.tol=.Machine\$double.eps</code> for maximal precision (however, this might be slow).
logml_iter	number of iterations (per chain) from the posterior distribution of d and τ . The samples are used for computing the marginal likelihood of the random-effects model with bridge sampling (if <code>logml="stan"</code>) and for obtaining parameter estimates (if <code>summarize="stan"</code>). Note that the argument <code>iter=2000</code> controls the number of iterations for estimation of the random-effect parameters per study in random-effects meta-analysis.
iter	number of MCMC iterations for the random-effects meta-analysis. Needs to be larger than usual to estimate the probability of all random effects being ordered (i.e., positive or negative).
silent_stan	whether to suppress the Stan progress bar.
...	further arguments passed to <code>rstan::sampling</code> (see stanmodel-method-sampling). Relevant MCMC settings concern the number of warmup samples that are discarded (<code>warmup=500</code>), the total number of iterations per chain (<code>iter=2000</code>), the number of MCMC chains (<code>chains=4</code>), whether multiple cores should be used (<code>cores=4</code>), and control arguments that make the sampling in Stan more robust, for instance: <code>control=list(adapt_delta=.97)</code> .

Details

Usually, in random-effects meta-analysis, the study-specific random-effects are allowed to be both negative or positive even when the prior on the overall effect size d is truncated to be positive). In contrast, the function `meta_ordered` fits and tests a model in which the random effects are forced to be either all positive or all negative. The direction of the study-specific random-effects is defined via the prior on the mode of the truncated normal distribution d . For instance, `d=prior("norm", c(0, .5), lower=0)` means that all random-effects are positive (not just the overall mean effect size).

The posterior summary statistics of the overall effect size in the model `ordered` refer to the the *average/mean* of the study-specific effect sizes (as implied by the fitted truncated normal distribution) and *not* to the location parameter d of the truncated normal distribution (which is only the mode, not the expected value of a truncated normal distribution).

The Bayes factor for the order-constrained model is computed using the encompassing Bayes factor. Since many posterior samples are required for this approach, the default number of MCMC iterations for `meta_ordered` is `iter=5000` per chain.

References

Haaf, J. M., & Rouder, J. N. (in press). Some do and some don't? Accounting for variability of individual difference structures. *Psychonomic Bulletin & Review*. doi: 10.3758/s13423-018-1522-x

See Also

[meta_bma](#), [meta_random](#)

Examples

```
### Bayesian Meta-Analysis with Order Constraints
# Note: The following code optimizes speed (for CRAN checks).
#       The settings are not suitable for actual data analysis!

data(towels)
set.seed(123)
mo <- meta_ordered(logOR, SE, study, towels,
                   d = prior("norm", c(mean=0, sd=.3), lower=0),
                   rel.tol=.01, iter=1000)

mo
plot_posterior(mo)
```

 meta_random

Bayesian Random-Effects Meta-Analysis

Description

Bayesian meta-analysis assuming that the effect size d varies across studies with standard deviation τ (i.e., a random-effects model).

Usage

```
meta_random(
  y,
  SE,
  labels,
  data,
  d = prior("norm", c(mean = 0, sd = 0.3)),
  tau = prior("invgamma", c(shape = 1, scale = 0.15)),
  rscale_contin = 1/2,
  rscale_discrete = sqrt(2)/2,
  centering = TRUE,
  logml = "integrate",
  summarize = "stan",
  ci = 0.95,
  rel.tol = .Machine$double.eps^0.3,
  logml_iter = 5000,
  silent_stan = TRUE,
  ...
)
```

Arguments

y effect size per study. Can be provided as (1) a numeric vector, (2) the quoted or unquoted name of the variable in data, or (3) a [formula](#) to include discrete or continuous moderator variables.

SE	standard error of effect size for each study. Can be a numeric vector or the quoted or unquoted name of the variable in data
labels	optional: character values with study labels. Can be a character vector or the quoted or unquoted name of the variable in data
data	data frame containing the variables for effect size y , standard error SE, labels, and moderators per study.
d	prior distribution on the average effect size d . The prior probability density function is defined via prior .
tau	prior distribution on the between-study heterogeneity tau (i.e., the standard deviation of the study effect sizes d_{study} in a random-effects meta-analysis. A (nonnegative) prior probability density function is defined via prior .
rscale_contin	scale parameter of the JZS prior for the continuous covariates.
rscale_discrete	scale parameter of the JZS prior for discrete moderators.
centering	whether continuous moderators are centered.
logml	how to estimate the log-marginal likelihood: either by numerical integration ("integrate") or by bridge sampling using MCMC/Stan samples ("stan"). To obtain high precision with <code>logml="stan"</code> , many MCMC samples are required (e.g., <code>logml_iter=10000</code> , <code>warmup=1000</code>).
summarize	how to estimate parameter summaries (mean, median, SD, etc.): Either by numerical integration (<code>summarize = "integrate"</code>) or based on MCMC/Stan samples (<code>summarize = "stan"</code>).
ci	probability for the credibility/highest-density intervals.
rel.tol	relative tolerance used for numerical integration using integrate . Use <code>rel.tol=.Machine\$double.eps</code> for maximal precision (however, this might be slow).
logml_iter	number of iterations (per chain) from the posterior distribution of d and τ . The samples are used for computing the marginal likelihood of the random-effects model with bridge sampling (if <code>logml="stan"</code>) and for obtaining parameter estimates (if <code>summarize="stan"</code>). Note that the argument <code>iter=2000</code> controls the number of iterations for estimation of the random-effect parameters per study in random-effects meta-analysis.
silent_stan	whether to suppress the Stan progress bar.
...	further arguments passed to <code>rstan::sampling</code> (see stanmodel-method-sampling). Relevant MCMC settings concern the number of warmup samples that are discarded (<code>warmup=500</code>), the total number of iterations per chain (<code>iter=2000</code>), the number of MCMC chains (<code>chains=4</code>), whether multiple cores should be used (<code>cores=4</code>), and control arguments that make the sampling in Stan more robust, for instance: <code>control=list(adapt_delta=.97)</code> .

Examples

```
### Bayesian Random-Effects Meta-Analysis
# Note: The following code optimizes speed (for CRAN checks).
#       The settings are not suitable for actual data analysis!
```



```

data(towels)
set.seed(123)
mr <- meta_random(logOR, SE, study, data = towels,
                  d = prior("norm", c(mean=0, sd=.3), lower = 0),
                  tau = prior("invgamma", c(shape = 1, scale = 0.15)),
                  rel.tol=.Machine$double.eps^.15, iter=1000)

mr
plot_posterior(mr)

```

plot.meta_pred *Plot Predicted Bayes Factors*

Description

Plot Predicted Bayes Factors

Usage

```

## S3 method for class 'meta_pred'
plot(x, which = "d_10_averaged", scale = "BF", ...)

```

Arguments

x	an object of the class "prediction" which contains observed and predicted Bayes factors
which	a character value defining which Bayes factor to plot (one of "d_10_fixed", "d_10_random", "d_10_averaged", "H1_fixed_vs_random")
scale	either plots Bayes factors ("BF"), inverse Bayes factors ("1/BF"), log Bayes factors ("log"), or the log-inverse Bayes factor ("1/log")
...	arguments passed to plot

plot.prior *Plot Prior Distribution*

Description

Plot the probability density function of a prior distribution.

Usage

```

## S3 method for class 'prior'
plot(x, from, to, ...)

```

Arguments

x	prior probability density function defined via prior .
from	lower boundary
to	upper boundary
...	further arguments passed to plot

Examples

```
p1 <- prior("t", c(location=0, scale=0.707, nu=1), 0, 3)
plot(p1, 0, 2)

# define custom prior pdf up to a constant:
p2 <- prior("custom", function(x) x^.5, 0, .5)
plot(p2)
```

plot_default

Plot Default Priors

Description

Plots default priors for the mean effect d and the standard deviation of effects τ .

Usage

```
plot_default(field = "psychology", effect = "d", ...)
```

Arguments

field	either "psychology" or "medicine" (uses partial matching, so "p" and "m" are sufficient)
effect	the type of effect size: either Cohen's d ("d"), Pearson correlations ("r"), Fisher's z -transformed correlations ("z"), or log-odds ratios ("logOR").
...	further arguments passed to plot (e.g., from, to)

See Also

[meta_default](#) for details on standard priors.

Examples

```
plot_default("psychology", "ttest", 0, 2)
plot_default("medicine", "logOR", 0, 2)
```

`plot_forest`*Forest Plot for Meta-Analysis*

Description

Plots estimated effect sizes for all studies.

Usage

```
plot_forest(  
  meta,  
  from,  
  to,  
  shrunked = "random",  
  summary = c("mean", "hpd"),  
  mar = c(4.5, 12, 4, 0.3),  
  cex.axis = 1,  
  ...  
)
```

Arguments

<code>meta</code>	fitted meta-analysis model
<code>from</code>	lower limit of the x-axis
<code>to</code>	upper limit of the x-axis
<code>shrunked</code>	which meta-analysis model should be used to show (shrunked) estimates of the study effect sizes. The name must match the corresponding name in the list <code>meta</code> . Can be suppressed by <code>shrunked = ""</code>
<code>summary</code>	character vector with two values: first, either "mean" or "50%"; and second, either highest-probability-density interval "hpd" or the Bayesian credibility interval "bci".
<code>mar</code>	margin of the plot in the order <code>c(bottom, left, top, right)</code> (see par)
<code>cex.axis</code>	size of the y-axis annotation for the labels of studies.
<code>...</code>	arguments passed to plot (e.g., <code>from</code> , <code>to</code>)

See Also

[meta_bma](#), [meta_fixed](#), [meta_random](#)

Examples

```
data(towels)  
mf <- meta_fixed(logOR, SE, study, towels)  
plot_forest(mf, mar = c(4.5,20,4,.2), xlab="Log Odds Ratio")
```

plot_posterior *Plot Posterior Distribution*

Description

Plot Posterior Distribution

Usage

```
plot_posterior(
  meta,
  parameter = "d",
  from,
  to,
  summary = c("mean", "hpd"),
  ...
)
```

Arguments

meta	fitted meta-analysis model
parameter	only for random-effects model: whether to plot "d" or "tau"
from	lower limit of the x-axis
to	upper limit of the x-axis
summary	character vector with two values: first, either "mean" or "50%"; and second, either highest-probability-density interval "hpd" or the Bayesian credibility interval "bci".
...	arguments passed to plot

See Also

[meta_bma](#), [meta_fixed](#), [meta_random](#)

power_pose *Data Set: Power Pose Effect*

Description

Includes six pre-registered replication studies testing whether participants feel more powerful if they adopt expansive as opposed to constrictive body postures. In the data set `power_pose_unfamiliar`, only those participants are included who were unfamiliar with the power pose effect.

Usage

power_pose

power_pose_unfamiliar

Format

A data frame with three variables:

study Authors of original study

n_high_power number of participants in high-power condition

n_low_power number of participants in low-power condition

mean_high_power mean rating in high-power condition on a 5-point Likert scale

mean_low_power mean rating in low-power condition on a 5-point Likert scale

sd_high_power standard deviation of ratings in high-power condition

sd_low_power standard deviation of ratings in low-power condition

t_value t-value for two-sample t-test

df degrees of freedom for two-sample t-test

two_sided_p_value two-sided p-value of two-sample t-test

one_sided_p_value one-sided p-value of two-sample t-test

effectSize Cohen's d, the standardized effect size (high vs. low power)

SE Standard error of Cohen's d

Data frame with 6 rows and 13 variables

An object of class `data.frame` with 6 rows and 13 columns.

Details

See Carney, Cuddy, and Yap (2010) for more details.

References

Carney, D. R., Cuddy, A. J. C., & Yap, A. J. (2010). Power posing: Brief nonverbal displays affect neuroendocrine levels and risk tolerance. *Psychological Science*, 21, 1363–1368.

Gronau, Q. F., Erp, S. V., Heck, D. W., Cesario, J., Jonas, K. J., & Wagenmakers, E.-J. (2017). A Bayesian model-averaged meta-analysis of the power pose effect with informed and default priors: the case of felt power. *Comprehensive Results in Social Psychology*, 2(1), 123-138. <https://doi.org/10.1080/23743603.2017.1326760>

Examples

```

data(power_pose)
head(power_pose)

# Simple fixed-effects meta-analysis
mfix <- meta_fixed(effectSize, SE, study,
                  data = power_pose)

mfix
plot_posterior(mfix)

```

predicted_bf	<i>Predicted Bayes Factors for a New Study</i>
--------------	--

Description

How much can be learned by an additional study? To judge this, this function samples the distribution of predicted Bayes factors for a new study given the current evidence.

Usage

```
predicted_bf(meta, SE, sample = 100, ...)
```

Arguments

meta	model-averaged meta-analysis (fitted with meta_bma).
SE	a scalar: the expected standard error of future study. For instance, $SE = 1/\sqrt{N}$ for standardized effect sizes and $N = \text{sample size}$
sample	number of simulated Bayes factors
...	further arguments passed to <code>rstan::sampling</code> to draw posterior samples for d and τ .

prior	<i>Prior Distribution</i>
-------	---------------------------

Description

Defines a prior distribution/probability density function for the average effect size d or for the heterogeneity of effect sizes τ .

Usage

```
prior(  
  family,  
  param,  
  lower,  
  upper,  
  label = "d",  
  rel.tol = .Machine$double.eps^0.5  
)
```

Arguments

family	a character value defining the distribution family.
param	numeric parameters for the distribution. See details for the definition of the parameters of each family.
lower	lower boundary for truncation of prior density. If family="beta", the interval [0,1] is rescaled to the interval [lower,upper]. Must be specified if family="custom".
upper	See lower.
label	optional: parameter label.
rel.tol	relative tolerance used for integrating the density of family="custom".

Details

The following prior distributions are currently implemented:

- "norm": Normal distribution with param = c(mean, sd) (see [Normal](#)).
- "t": Student t distribution with param = c(location, scale, nu) (see [dist.Student.t](#)). Note that a Cauchy distribution is defined by setting the degrees of freedom nu=1.
- "invgamma": Inverse gamma distribution with param = c(shape, scale) (see [dist.Inverse.Gamma](#)).
- "beta": (Scaled) beta distribution with param = c(shape1, shape2) (see [Beta](#)).
- "custom": User-specified prior density function defined by param (see examples; the density must be nonnegative and vectorized, but is normalized internally). Integration is performed from (-Inf, Inf), which requires that the function returns zeros (and not NAs) for values not in the support of the distribution.

Value

an object of the class prior: a density function with the arguments x (parameter values) and log (whether to return density or log-density).

Examples

```
### Half-Normal Distribution  
p1 <- prior("norm", c(mean=0, sd=.3), lower = 0)  
p1  
p1(c(-1, 1, 3))
```

```
plot(p1, -.1, 1)

### Half-Cauchy Distribution
p2 <- prior("t", c(location = 0, scale = .3, nu = 1), lower = 0)
plot(p2, -.5, 3)

### Custom Prior Distribution
p3 <- prior("custom", function(x) x^2, 0, 1)
plot(p3, -.1, 1.2)
```

towels

Data Set: Reuse of Towels in Hotels

Description

Set of studies that investigated whether people reuse towels in hotels more often if they are provided with a descriptive norm (Scheibehenne, Jamil, & Wagenmakers, 2016).

Usage

```
towels
```

Format

A data frame with three variables:

study Authors of original study (see Scheibehenne et. al, 2016)

logOR Measure of effect size: log-odds ratio of towel reuse (descriptive-social-norm vs. control)

SE Measure of precision: standard error of log-odds ratio per study

Details

Two groups of hotel guests received different messages that encouraged them to reuse their towels. One message simply informed the guests about the benefits of environmental protection (the control condition), and the other message indicated that the majority of guests actually reused their towels in the past (the descriptive-social-norm condition). The results suggested that the latter message facilitated towel reuse.

References

Scheibehenne, B., Jamil, T., & Wagenmakers, E.-J. (2016). Bayesian Evidence Synthesis Can Reconcile Seemingly Inconsistent Results: The Case of Hotel Towel Reuse. *Psychological Science*, 27(7), 1043–1046. <https://doi.org/10.1177/0956797616644081>

Examples

```
data(towels)
head(towels)
```


Index

*Topic **datasets**

facial_feedback, 5
power_pose, 20
towels, 24

Beta, 23
bma, 3, 4

dist.Inverse.Gamma, 23
dist.Student.t, 23

facial_feedback, 5
formula, 7, 9, 11, 13, 15

inclusion, 3, 6
integrate, 4, 8, 12, 14, 16

meta_bma, 3, 7, 10, 14, 19, 20, 22
meta_default, 9, 18
meta_fixed, 3, 4, 6, 9, 11, 19, 20
meta_ordered, 12
meta_random, 3, 4, 6, 9, 14, 15, 19, 20
metaBMA (metaBMA-package), 2
metaBMA-package, 2

Normal, 23

par, 19
plot, 17–20
plot.meta_pred, 17
plot.prior, 17
plot_default, 10, 18
plot_forest, 3, 19
plot_posterior, 3, 20
power_pose, 20
power_pose_unfamiliar (power_pose), 20
predicted_bf, 22
prior, 2, 3, 7, 12, 13, 16, 18, 22

towels, 24