Package 'jarbes'

December 5, 2024

Type Package

Title Just a Rather Bayesian Evidence Synthesis

Version 2.2.3

Date 2024-12-05

Depends R (>= 4.0.0)

Imports rjags, R2jags, stats, graphics, ggplot2, ggExtra, MASS, grid, gridExtra, mcmcplots, bookdown, tidyr, kableExtra, GGally, qpdf

SystemRequirements JAGS (>= 4.3.0) (see http://mcmc-jags.sourceforge.net)

Description Provides a new class of Bayesian meta-analysis models that incorporates a model for internal and external validity bias. In this way, it is possible to combine studies of diverse quality and different types. For example, we can combine the results of randomized control trials (RCTs) with the results of observational studies (OS).

License GPL (>= 2)

Repository CRAN

RoxygenNote 7.3.2

Encoding UTF-8

LazyData true

Author Pablo Emilio Verde [aut, cre]

Maintainer Pablo Emilio Verde <pabloemilio.verde@hhu.de>

NeedsCompilation no

Date/Publication 2024-12-05 16:50:07 UTC

Contents

`b3lmeta` *Bayesian Meta-Analysis for Combining Studies*

Description

This function performers a Bayesian meta-analysis

Usage

```
b3lmeta(
  data,
  mean.mu.0 = 0,
  sd.mu.0 = 10,
  scale.sigma.between = 0.5,
  df.scale.between = 1,
  scale.sigma.within = 0.5,
  df.scale.within = 1,
  nr.chains = 2,
  nr.iterations = 10000,
  nr.adapt = 1000,
  nr.burnin = 1000,
  nr.thin = 1
)
```

Arguments

`data`	A data frame with at least three columns with the following names: 1) TE = treatment effect, 2) seTE = the standard error of the treatment effect. 3) design = indicates study type or clustering subgroup.
`mean.mu.0`	Prior mean of the overall mean parameter mu.0 (mean across designs), default value is 0.
`sd.mu.0`	Prior standard deviation of mu.0 (mean across designs), the default value is 10.
`scale.sigma.between`	Prior scale parameter for scale gamma distribution for the precision between study types. The default value is 0.5.
`df.scale.between`	Degrees of freedom of the scale gamma distribution for the precision between study types. The default value is 1, which results in a Half Cauchy distribution for the standard deviation between studies. Larger values e.g. 30 corresponds to a Half Normal distribution.
`scale.sigma.within`	Prior scale parameter for scale gamma distribution for the precision within study types. The default value is 0.5.

`df.scale.within`
: Degrees of freedom of the scale gamma distribution for the precision within study types. The default value is 1, which results in a Half Cauchy distribution for the standard deviation between studies. Larger values e.g. 30 corresponds to a Half Normal distribution.

`nr.chains`
: Number of chains for the MCMC computations, default 2.

`nr.iterations`
: Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.

`nr.adapt`
: Number of iterations in the adaptation process, default is 1000. Some models may need more iterations during adptation.

`nr.burnin`
: Number of iteration discard for burn-in period, default is 1000. Some models may need a longer burnin period.

`nr.thin`
: Thinning rate, it must be a positive integer, the default value 1.

Details

The results of the object of the class bcmeta can be extracted with R2jags or with rjags. In addition a summary, a print and a plot functions are implemented for this type of object.

Value

This function returns an object of the class "bmeta". This object contains the MCMC output of each parameter and hyper-parameter in the model and the data frame used for fitting the model.

References

Verde, P.E. (2021) A Bias-Corrected Meta-Analysis Model for Combining Studies of Different Types and Quality. Biometrical Journal; 1–17.

Examples

```
## Not run:
library(jarbes)



## End(Not run)
```

`bcdpmeta` *Bias Corrected Meta-Analysis with Dirichlet Process Priors*

Description

This function performers a Bayesian meta-analysis with DP as random effects

Usage

```
bcdpmeta(
  data,
  mean.mu.0 = 0,
  sd.mu.0 = 10,
  scale.sigma.between = 0.5,
  df.scale.between = 1,
  B.lower = 0,
  B.upper = 10,
  a.0 = 1,
  a.1 = 1,
  alpha.0 = 0.03,
  alpha.1 = 10,
  K = 30,
  nr.chains = 2,
  nr.iterations = 10000,
  nr.adapt = 1000,
  nr.burnin = 1000,
  nr.thin = 1
)
```

Arguments

`data`	A data frame with at least two columns with the following names: 1) TE = treatment effect, 2) seTE = the standard error of the treatment effect.
`mean.mu.0`	Prior mean of the mean of the base distribution default value is mean.mu.0 = 0.
`sd.mu.0`	Prior standard deviation of the base distribution, the default value is 10.
`scale.sigma.between`	Prior scale parameter for scale gamma distribution for the precision between studies. The default value is 0.5.
`df.scale.between`	Degrees of freedom of the scale gamma distribution for the precision between studies. The default value is 1, which results in a Half Cauchy distribution for the standard deviation between studies. Larger values e.g. 30 corresponds to a Half Normal distribution.
`B.lower`	Lower bound of the bias parameter B, the default value is 0.
`B.upper`	Upper bound of the bias parameter B, the default value is 10.
`a.0`	Parameter for the prior Beta distribution for the probability of bias. Default value is a0 = 1.
`a.1`	Parameter for the prior Beta distribution for the probability of bias. Default value is a1 = 1.
`alpha.0`	Lower bound of the uniform prior for the concentration parameter for the DPM, default value is alpha.0 = 0.03.
`alpha.1`	Upper bound of the uniform prior for the concentration parameter for the DPM, default value is alpha.1 = 10.

K	Maximum number of clusters in the DPM, default value is K = 30.
nr.chains	Number of chains for the MCMC computations, default 2.
nr.iterations	Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.
nr.adapt	Number of iterations in the adaptation process, default is 1000. Some models may need more iterations during adptation.
nr.burnin	Number of iteration discard for burn-in period, default is 1000. Some models may need a longer burnin period.
nr.thin	Thinning rate, it must be a positive integer, the default value 1.

Details

The results of the object of the class bcdpmeta can be extracted with R2jags or with rjags. In addition a summary, a print and a plot functions are implemented for this type of object.

Value

This function returns an object of the class "bcdpmeta". This object contains the MCMC output of each parameter and hyper-parameter in the model and the data frame used for fitting the model.

References

Verde, P.E. (2021) A Bias-Corrected Meta-Analysis Model for Combining Studies of Different Types and Quality. Biometrical Journal; 1–17.

Examples

```
## Not run:
library(jarbes)


# Example: Stemcells

data("stemcells")
stemcells$TE = stemcells$effect.size
stemcells$seTE = stemcells$se.effect

bm1 = bcdpmeta(stemcells)
summary(bm1)


## End(Not run)
```

bcmeta *Bias-Corrected Meta-Analysis for Combining Studies of Different Types and Quality*

Description

This function performers a Bayesian meta-analysis to jointly combine different types of studies. The random-effects follows a finite mixture of normal distributions.

Usage

```
bcmeta(
  data,
  mean.mu = 0,
  sd.mu = 10,
  scale.sigma.between = 0.5,
  df.scale.between = 1,
  B.lower = 0,
  B.upper = 10,
  a.0 = 1,
  a.1 = 1,
  nu = 0.5,
  nu.estimate = FALSE,
  b.0 = 1,
  b.1 = 2,
  nr.chains = 2,
  nr.iterations = 10000,
  nr.adapt = 1000,
  nr.burnin = 1000,
  nr.thin = 1
)
```

Arguments

`data`	A data frame with at least two columns with the following names: 1) TE = treatment effect, 2) seTE = the standard error of the treatment effect.
`mean.mu`	Prior mean of the overall mean parameter mu, default value is 0.
`sd.mu`	Prior standard deviation of mu, the default value is 10.
`scale.sigma.between`	Prior scale parameter for scale gamma distribution for the precision between studies. The default value is 0.5.
`df.scale.between`	Degrees of freedom of the scale gamma distribution for the precision between studies. The default value is 1, which results in a Half Cauchy distribution for the standard deviation between studies. Larger values e.g. 30 corresponds to a Half Normal distribution.

B.lower	Lower bound of the bias parameter B, the default value is 0.
B.upper	Upper bound of the bias parameter B, the default value is 10.
a.0	Parameter for the prior Beta distribution for the probability of bias. Default value is a0 = 1.
a.1	Parameter for the prior Beta distribution for the probability of bias. Default value is a1 = 1.
nu	Parameter for the Beta distribution for the quality weights. The default value is nu = 0.5.
nu.estimate	If TRUE, then we estimate nu from the data.
b.0	If nu.estimate = TRUE, this parameter is the shape parameter of the prior Gamma distribution for nu.
b.1	If nu.estimate = TRUE, this parameter is the rate parameter of the prior Gamma distribution for nu. Note that E(nu) = b.0/b.1 and we need to choose b.0 « b.1.
nr.chains	Number of chains for the MCMC computations, default 2.
nr.iterations	Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.
nr.adapt	Number of iterations in the adaptation process, defualt is 1000. Some models may need more iterations during adptation.
nr.burnin	Number of iteration discared for burnin period, default is 1000. Some models may need a longer burnin period.
nr.thin	Thinning rate, it must be a positive integer, the default value 1.

Details

The results of the object of the class bcmeta can be extracted with R2jags or with rjags. In addition a summary, a print and a plot functions are implemented for this type of object.

Value

This function returns an object of the class "bcmeta". This object contains the MCMC output of each parameter and hyper-parameter in the model and the data frame used for fitting the model.

References

Verde, P. E. (2017) Two Examples of Bayesian Evidence Synthesis with the Hierarchical Meta-Regression Approach. Chap.9, pag 189-206. Bayesian Inference, ed. Tejedor, Javier Prieto. InTech.

Verde, P.E. (2021) A Bias-Corrected Meta-Analysis Model for Combining Studies of Different Types and Quality. Biometrical Journal; 1–17.

Examples

```
## Not run:
library(jarbes)

# Example ppvipd data
```

```
data(ppvipd)


## End(Not run)
```

`bcmixmeta` *Bias Corrected Meta-Analysis with Dirichlet Mixture Process Priors for the biased component*

Description

This function performers a Bayesian meta-analysis with DPM as random effects

Usage

```
bcmixmeta(
  data,
  x = NULL,
  mean.mu.0 = 0,
  sd.mu.0 = 10,
  scale.sigma.between = 0.5,
  df.scale.between = 1,
  scale.sigma.beta = 0.5,
  df.scale.beta = 1,
  B.lower = -15,
  B.upper = 15,
  a.0 = 0.5,
  a.1 = 1,
  alpha.0 = 0.03,
  alpha.1 = 2,
  K = 10,
  bilateral.bias = FALSE,
  nr.chains = 2,
  nr.iterations = 10000,
  nr.adapt = 1000,
  nr.burnin = 1000,
  nr.thin = 1
)
```

Arguments

`data`	A data frame with at least two columns with the following names: 1) TE = treatment effect, 2) seTE = the standard error of the treatment effect.
`x`	a covariate to perform meta-regression.
`mean.mu.0`	Prior mean of the mean of the base distribution default value is mean.mu.0 = 0.

`sd.mu.0` Prior standard deviation of the base distribution, the default value is 10^-6.

`scale.sigma.between`
Prior scale parameter for scale gamma distribution for the precision between studies. The default value is 0.5.

`df.scale.between`
Degrees of freedom of the scale gamma distribution for the precision between studies. The default value is 1, which results in a Half Cauchy distribution for the standard deviation between studies. Larger values e.g. 30 corresponds to a Half Normal distribution.

`scale.sigma.beta`
Prior scale parameter for the scale.gamma distribution for the precision between study biases.

`df.scale.beta` Degrees of freedom of the scale gamma distribution for the precision between study biases. The default value is 1, which results in a Half Cauchy distribution for the standard deviation between biases.

`B.lower` Lower bound of the bias parameter B, the default value is -15.

`B.upper` Upper bound of the bias parameter B, the default value is 15.

`a.0` Parameter for the prior Beta distribution for the probability of bias. Default value is a0 = 0.5.

`a.1` Parameter for the prior Beta distribution for the probability of bias. Default value is a1 = 1.

`alpha.0` Lower bound of the uniform prior for the concentration parameter for the DP, the default value is 0.5.

`alpha.1` Upper bound of the uniform prior for the concentration parameter for the DP, the default value depends on the sample size, see the example below. We give as working value alpha.1 = 2

`K` Maximum number of clusters in the DP, the default value depends on alpha.1, see the example below. We give as working value K = 10.

`bilateral.bias` Experimental option, which indicates if bias could be to the left and to the right of the model of interest. If bilateral.bias==TRUE, then the function generates three mean and sorts the means in two groups: mean_bias_left, mean_theta, mean_bias_right.

`nr.chains` Number of chains for the MCMC computations, default 2.

`nr.iterations` Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.

`nr.adapt` Number of iterations in the adaptation process, default is 1000. Some models may need more iterations during adptation.

`nr.burnin` Number of iteration discard for burn-in period, default is 1000. Some models may need a longer burnin period.

`nr.thin` Thinning rate, it must be a positive integer, the default value 1.

Details

The results of the object of the class bcmixmeta can be extracted with R2jags or with rjags. In addition a summary, a print and a plot functions are implemented for this type of object.

Value

This function returns an object of the class "bcmixmeta". This object contains the MCMC output of each parameter and hyper-parameter in the model and the data frame used for fitting the model.

References

Verde, P.E. and Rosner, G. L. (2024) A Bias-Corrected Bayesian Nonparamteric Model for Combining Studies with Varying Quality in Meta-Analysis. Biometrical Journal; (under revision).

Examples

```
## Not run:
library(jarbes)


# Example: Stemcells

data("stemcells")
stemcells$TE = stemcells$effect.size
stemcells$seTE = stemcells$se.effect

# Beta(0.5, 1)
a.0 = 0.5
a.1 = 1

# alpha.max
 N = dim(stemcells)[1]
 alpha.max = 1/5 *((N-1)*a.0 - a.1)/(a.0 + a.1)

alpha.max

# K.max
K.max = 1 + 5*alpha.max
K.max = round(K.max)

K.max

set.seed(20233)

bcmix.2.stemcell = bcmixmeta(stemcells,
                             mean.mu.0=0, sd.mu.0=100,
                             B.lower = -15,
                             B.upper = 15,
                             alpha.0 = 0.5,
                             alpha.1 = alpha.max,
                             a.0 = a.0,
                             a.1 = a.1,
                             K = K.max,
                             sort.priors = FALSE,
                             df.scale.between = 1,
                             scale.sigma.between = 0.5,
                             nr.chains = 4,
```

```
                              nr.iterations = 50000,
                              nr.adapt = 1000,
                              nr.burnin = 10000,
                              nr.thin = 4)


 diagnostic(bcmix.2.stemcell, y.lim = c(-1, 15), title.plot = "Default priors")


 bcmix.2.stemcell.mcmc <- as.mcmc(bcmix.1.stemcell$BUGSoutput$sims.matrix)


theta.names <- paste(paste("theta[",1:31, sep=""),"]", sep="")
theta.b.names <- paste(paste("theta.bias[",1:31, sep=""),"]", sep="")

theta.b.greek.names <- paste(paste("theta[",1:31, sep=""),"]^B", sep="")
theta.greek.names <- paste(paste("theta[",1:31, sep=""),"]", sep="")


caterplot(bcmix.2.stemcell.mcmc,
         parms = theta.names,               # theta
         labels = theta.greek.names,
         greek = T,
         labels.loc="axis", cex =0.7,
         col = "black",
         style = "plain",
         reorder = F,
         val.lim =c(-6, 16),
         quantiles = list(outer=c(0.05,0.95),inner=c(0.16,0.84)),
         x.lab = "Effect: mean difference"
)
title( "95% posterior intervals of studies' effects")
caterplot(bcmix.2.stemcell.mcmc,
         parms = theta.b.names,             # theta.bias
         labels = theta.greek.names,
         greek = T,
         labels.loc="no",
         cex = 0.7,
         col = "grey",
         style = "plain", reorder = F,
         val.lim =c(-6, 16),
         quantiles = list(outer=c(0.025,0.975),inner=c(0.16,0.84)),
         add = TRUE,
         collapse=TRUE, cat.shift= -0.5,
)

attach.jags(bcmix.2.stemcell, overwrite = TRUE)
abline(v=mean(mu.0), lwd =2, lty =2)

legend(9, 20, legend = c("bias corrected", "biased"),
     lty = c(1,1), lwd = c(2,2), col = c("black", "grey"))
```

```
## End(Not run)
```

---

bmeta *Bayesian Meta-Analysis for Combining Studies*

Description

This function performers a Bayesian meta-analysis

Usage

```
bmeta(
  data,
  mean.mu = 0,
  sd.mu = 10,
  scale.sigma.between = 0.5,
  df.scale.between = 1,
  nr.chains = 2,
  nr.iterations = 10000,
  nr.adapt = 1000,
  nr.burnin = 1000,
  nr.thin = 1,
  be.quiet = FALSE
)
```

Arguments

`data`	A data frame with at least two columns with the following names: 1) TE = treatment effect, 2) seTE = the standard error of the treatment effect.
`mean.mu`	Prior mean of the overall mean parameter mu, default value is 0.
`sd.mu`	Prior standard deviation of mu, the default value is 10.
`scale.sigma.between`	Prior scale parameter for scale gamma distribution for the precision between studies. The default value is 0.5.
`df.scale.between`	Degrees of freedom of the scale gamma distribution for the precision between studies. The default value is 1, which results in a Half Cauchy distribution for the standard deviation between studies. Larger values e.g. 30 corresponds to a Half Normal distribution.
`nr.chains`	Number of chains for the MCMC computations, default 2.
`nr.iterations`	Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.
`nr.adapt`	Number of iterations in the adaptation process, default is 1000. Some models may need more iterations during adptation.

nr.burnin	Number of iteration discard for burn-in period, default is 1000. Some models may need a longer burnin period.
nr.thin	Thinning rate, it must be a positive integer, the default value 1.
be.quiet	Do not print warning message if the model does not adapt. The default value is FALSE. If you are not sure about the adaptation period choose be.quiet=TRUE.

Details

The results of the object of the class bcmeta can be extracted with R2jags or with rjags. In addition a summary, a print and a plot functions are implemented for this type of object.

Value

This function returns an object of the class "bmeta". This object contains the MCMC output of each parameter and hyper-parameter in the model and the data frame used for fitting the model.

References

Verde, P.E. (2021) A Bias-Corrected Meta-Analysis Model for Combining Studies of Different Types and Quality. Biometrical Journal; 1–17.

Examples

```
## Not run:
library(jarbes)

#Example: ppvipd

data(ppvipd)
bm1 = bmeta(ppvipd)

summary(bm1)
plot(bm1, x.lim = c(-3, 1), y.lim = c(0, 3))

diagnostic(bm1, study.names = ppvipd$name, post.p.value.cut = 0.1,
           lwd.forest = 1, shape.forest = 4)

# Example: Stemcells

data("stemcells")
stemcells$TE = stemcells$effect.size
stemcells$seTE = stemcells$se.effect

bm2 = bmeta(stemcells)
summary(bm2)
plot(bm2, x.lim = c(-1, 7), y.lim = c(0, 1))

diagnostic(bm2, study.names = stemcells$trial,
           post.p.value.cut = 0.05,
           lwd.forest = 0.5, shape.forest = 4)
```

```
diagnostic(bm2, post.p.value.cut = 0.05,
           lwd.forest = 0.5, shape.forest = 4)

## End(Not run)
```

`covid19` *Meta-analysis: Observational studies assessing the impact of risk factors on the severity and mortality of COVID-19 cases*

Description

Meta-analysis of 40 Observational Studies from PubMed, Cocharane Library and SciELO databases that assessed the impact of diabetes, hypertension, cardiovascular disease, and the use of ACEI/ARB on severity and mortality of COVID-19 cases.

Format

A dataframe with 89 rows and 12 columns. Each row represents study results, the columns are:

author Principal author and year of publication.

endpoint Endoint: severity or mortality.

risk.factor Possible risk factors: diabetes, hypertension, cardiovascular, ACE_ARB.

event.e Number of events in the group with risk factor.

n.e Number of patients in the group with risk factor.

event.c Number of events in the group without risk factor.

n.c Number of patients in the group with risk factor.

design Study design: Case Series, Cross Sectional and Retrospective Cohort.

TE Log Odds Ratio

seTE Standard Error of the Log Odds Ratio

logitPc Logit transformation of the proportion of events in the control group.

N Total number of patients.

Source

de Almeida-Pititto, B., Dualib, P.M., Zajdenverg, L. et al. Severity and mortality of COVID 19 in patients with diabetes, hypertension and cardiovascular disease: a meta-analysis. Diabetol Metab Syndr 12, 75 (2020). https://doi.org/10.1186/s13098-020-00586-4

diagnostic *Generic diagnostic function.*

Description

Generic diagnostic function.

Usage

```
diagnostic(object, ...)
```

Arguments

object	The object generated by the function hmr.
...	...

diagnostic.b3lmeta *Diagnostic function for b3lmeta object in jarbes*

Description

This function performers an approximated Bayesian cross-validation for a b3lmeta object

Usage

```
## S3 method for class 'b3lmeta'
diagnostic(
  object,
  post.p.value.cut = 0.05,
  study.names = NULL,
  size.forest = 0.4,
  lwd.forest = 0.2,
  shape.forest = 23,
  ...
)
```

Arguments

object	The object generated by the function b3lmeta.
post.p.value.cut	Posterior p-value cut point to assess outliers.
study.names	Character vector containing names of the studies used.
size.forest	Size of the center symbol mark in the forest-plot lines
lwd.forest	Thickness of the lines in the forest-plot
shape.forest	Type of symbol for the center mark in the forest-plot lines
...	...

`diagnostic.bcdpmeta` *Diagnostic function for bcdpmeta object in jarbes*

Description

This function performers an approximated Bayesian cross-validation for a bcmeta object and specially designed diagnostics to detect the existence of a biased component.

Usage

```
## S3 method for class 'bcdpmeta'
diagnostic(
  object,
  post.p.value.cut = 0.05,
  study.names = NULL,
  size.forest = 0.4,
  lwd.forest = 0.2,
  shape.forest = 23,
  bias.plot = TRUE,
  cross.val.plot = FALSE,
  level = c(0.5, 0.75, 0.95),
  x.lim = c(0, 1),
  y.lim = c(0, 10),
  x.lab = "P(Bias)",
  y.lab = "Mean Bias",
  title.plot = paste("Bias Diagnostics Contours (50%, 75% and 95%)"),
  kde2d.n = 25,
  marginals = TRUE,
  bin.hist = 30,
  color.line = "black",
  color.hist = "white",
  color.data.points = "black",
  alpha.data.points = 0.1,
  S = 5000,
  ...
)
```

Arguments

`object`	The object generated by the function b3lmeta.
`post.p.value.cut`	Posterior p-value cut point to assess outliers.
`study.names`	Character vector containing names of the studies used.
`size.forest`	Size of the center symbol mark in the forest-plot lines
`lwd.forest`	Thickness of the lines in the forest-plot
`shape.forest`	Type of symbol for the center mark in the forest-plot lines

bias.plot	Display the bias plot. The default is TRUE.
cross.val.plot	Display the cross validation plot. The default is FALSE.
level	Vector with the probability levels of the contour plot. The default values are: 0.5, 0.75, and 0.95.
x.lim	Numeric vector of length 2 specifying the x-axis limits.
y.lim	Numeric vector of length 2 specifying the y-axis limits.
x.lab	Text with the label of the x-axis.
y.lab	Text with the label of the y-axis.
title.plot	Text for setting a title in the bias plot.
kde2d.n	The number of grid points in each direction for the non-parametric density estimation. The default is 25.
marginals	If TRUE the marginal histograms of the posteriors are added to the plot.
bin.hist	The number of bins in for the histograms. The default value is 30.
color.line	The color of the contour lines. The default is "black.
color.hist	The color of the histogram bars. The default is "white".
color.data.points	The color of the data points. The default is "black".
alpha.data.points	Transparency of the data points.
S	The number of sample values from the joint posterior distribution used to approximate the contours. The default is S=5000.
...	...

`diagnostic.bcmeta` *Diagnostic function for bcmeta object in jarbes*

Description

This function performers an approximated Bayesian cross-validation for a bcmeta object and specially designed diagnostics to detect the existence of a biased component.

Usage

```
## S3 method for class 'bcmeta'
diagnostic(
  object,
  post.p.value.cut = 0.05,
  study.names = NULL,
  size.forest = 0.4,
  lwd.forest = 0.2,
  shape.forest = 23,
  bias.plot = TRUE,
```

```
  cross.val.plot = TRUE,
  level = c(0.5, 0.75, 0.95),
  x.lim = c(0, 1),
  y.lim = c(0, 10),
  x.lab = "P(Bias)",
  y.lab = "Mean Bias",
  title.plot = paste("Bias Diagnostics Contours (50%, 75% and 95%)"),
  kde2d.n = 25,
  marginals = TRUE,
  bin.hist = 30,
  color.line = "black",
  color.hist = "white",
  color.data.points = "black",
  alpha.data.points = 0.1,
  S = 5000,
  ...
)
```

Arguments

`object`	The object generated by the function b3lmeta.
`post.p.value.cut`	Posterior p-value cut point to assess outliers.
`study.names`	Character vector containing names of the studies used.
`size.forest`	Size of the center symbol mark in the forest-plot lines
`lwd.forest`	Thickness of the lines in the forest-plot
`shape.forest`	Type of symbol for the center mark in the forest-plot lines
`bias.plot`	Display the bias plot. The default is TRUE.
`cross.val.plot`	Display the cross validation plot. The default is TRUE.
`level`	Vector with the probability levels of the contour plot. The default values are: 0.5, 0.75, and 0.95.
`x.lim`	Numeric vector of length 2 specifying the x-axis limits.
`y.lim`	Numeric vector of length 2 specifying the y-axis limits.
`x.lab`	Text with the label of the x-axis.
`y.lab`	Text with the label of the y-axis.
`title.plot`	Text for setting a title in the bias plot.
`kde2d.n`	The number of grid points in each direction for the non-parametric density estimation. The default is 25.
`marginals`	If TRUE the marginal histograms of the posteriors are added to the plot.
`bin.hist`	The number of bins in for the histograms. The default value is 30.
`color.line`	The color of the contour lines. The default is "black.
`color.hist`	The color of the histogram bars. The default is "white".
`color.data.points`	The color of the data points. The default is "black".

alpha.data.points
: Transparency of the data points.

S
: The number of sample values from the joint posterior distribution used to approximate the contours. The default is S=5000.

...
: ...

`diagnostic.bcmixmeta` *Diagnostic function for bcmixmeta object in jarbes*

Description

This function performers an approximated Bayesian cross-validation for a bcmeta object and specially designed diagnostics to detect the existence of a biased component.

Usage

```
## S3 method for class 'bcmixmeta'
diagnostic(
  object,
  post.p.value.cut = 0.05,
  study.names = NULL,
  size.forest = 0.4,
  lwd.forest = 0.2,
  shape.forest = 23,
  bias.plot = TRUE,
  cross.val.plot = FALSE,
  level = c(0.5, 0.75, 0.95),
  x.lim = c(0, 1),
  y.lim = c(0, 10),
  x.lab = "P(Bias)",
  y.lab = "Mean Bias",
  title.plot = paste("Bias Diagnostics Contours (50%, 75% and 95%)"),
  kde2d.n = 25,
  marginals = TRUE,
  bin.hist = 30,
  color.line = "black",
  color.hist = "white",
  color.data.points = "black",
  alpha.data.points = 0.1,
  S = 5000,
  ...
)
```

Arguments

object
: The object generated by the function b3lmeta.

post.p.value.cut	Posterior p-value cut point to assess outliers.
study.names	Character vector containing names of the studies used.
size.forest	Size of the center symbol mark in the forest-plot lines
lwd.forest	Thickness of the lines in the forest-plot
shape.forest	Type of symbol for the center mark in the forest-plot lines
bias.plot	Display the bias plot. The default is TRUE.
cross.val.plot	Display the cross validation plot. The default is FALSE.
level	Vector with the probability levels of the contour plot. The default values are: 0.5, 0.75, and 0.95.
x.lim	Numeric vector of length 2 specifying the x-axis limits.
y.lim	Numeric vector of length 2 specifying the y-axis limits.
x.lab	Text with the label of the x-axis.
y.lab	Text with the label of the y-axis.
title.plot	Text for setting a title in the bias plot.
kde2d.n	The number of grid points in each direction for the non-parametric density estimation. The default is 25.
marginals	If TRUE the marginal histograms of the posteriors are added to the plot.
bin.hist	The number of bins in for the histograms. The default value is 30.
color.line	The color of the contour lines. The default is "black.
color.hist	The color of the histogram bars. The default is "white".
color.data.points	The color of the data points. The default is "black".
alpha.data.points	Transparency of the data points.
S	The number of sample values from the joint posterior distribution used to approximate the contours. The default is S=5000.
...	...

`diagnostic.bmeta` *Diagnostic function for bmeta object in jarbes*

Description

This function performers an approximated Bayesian cross-validation for a b3lmeta object

Usage

```
## S3 method for class 'bmeta'
diagnostic(
  object,
  post.p.value.cut = 0.05,
  median.w = 1.5,
  study.names = NULL,
  size.forest = 0.4,
  lwd.forest = 0.2,
  shape.forest = 23,
  ...
)
```

Arguments

object	The object generated by the function bmeta.
post.p.value.cut	Posterior p-value cut point to assess outliers.
median.w	Change color if median of a weight > median.w. The default value is 1.5.
study.names	Character vector containing names of the studies used.
size.forest	Size of the center symbol mark in the forest-plot lines
lwd.forest	Thickness of the lines in the forest-plot
shape.forest	Type of symbol for the center mark in the forest-plot lines
...	...

---

diagnostic.hmr *Diagnostic function for hmr object in jarbes*

Description

This function performers a specially designed diagnostic for a hmr object

Usage

```
## S3 method for class 'hmr'
diagnostic(
  object,
  median.w = 1.5,
  study.names,
  size.forest = 0.4,
  lwd.forest = 0.2,
  shape.forest = 23,
  mu.phi = TRUE,
  mu.phi.x.lim.low = -10,
```

```
  mu.phi.x.lim.up = 10,
  colour.hist.mu.phi = "royalblue",
  colour.prior.mu.phi = "black",
  colour.posterior.mu.phi = "blue",
  title.plot.mu.phi = "Prior-to-Posterior Sensitivity",
  title.plot.weights = "Outlier Detection",
  ...
)
```

Arguments

`object`	The object generated by the function hmr.
`median.w`	Change colour if median of a weight > median.w. The default value is 1.5.
`study.names`	Character vector containing names of the studies used.
`size.forest`	Size of the center symbol mark in the forest-plot lines
`lwd.forest`	Thickness of the lines in the forest-plot
`shape.forest`	Type of symbol for the center mark in the forest-plot lines
`mu.phi`	Prior-to-posterior sensitivity analysis of mu.phi. Default value is TRUE.
`mu.phi.x.lim.low`	Lower limit of the prior to posterior plot for mu.phi
`mu.phi.x.lim.up`	Upper limit of the prior to posterior plot for mu.phi
`colour.hist.mu.phi`	colour of the posterior mu.phi histogram
`colour.prior.mu.phi`	colour of the prior of mu.phi
`colour.posterior.mu.phi`	colour of the posterior of mu.phi
`title.plot.mu.phi`	Text for the title in the mu phi plot.
`title.plot.weights`	Text for the title of the posterior weights.
`...`	...

`diagnostic.metarisk` *Diagnostic function for metarisk object in jarbes*

Description

This function performers a specially designed diagnostic for a metarisk object

Usage

```
## S3 method for class 'metarisk'
diagnostic(
  object,
  median.w = 1.5,
  study.names,
  size.forest = 0.4,
  lwd.forest = 0.2,
  shape.forest = 23,
  ...
)
```

Arguments

object	The object generated by the function hmr.
median.w	Change color if median of a weight > median.w. The default value is 1.5.
study.names	Character vector containing names of the studies used.
size.forest	Size of the center symbol mark in the forest-plot lines
lwd.forest	Thickness of the lines in the forest-plot
shape.forest	Type of symbol for the center mark in the forest-plot lines
...	...

---

dpmeta *Bayesian Meta-Analysis with Dirichlet Process Priors*

Description

This function performers a Bayesian meta-analysis with DP as random effects

Usage

```
dpmeta(
  data,
  mean.mu.0 = 0,
  sd.mu.0 = 10,
  scale.sigma.between = 0.5,
  df.scale.between = 1,
  alpha.0 = 0.03,
  alpha.1 = 10,
  K = 30,
  nr.chains = 2,
  nr.iterations = 10000,
  nr.adapt = 1000,
  nr.burnin = 1000,
  nr.thin = 1
)
```

Arguments

`data`	A data frame with at least two columns with the following names: 1) TE = treatment effect, 2) seTE = the standard error of the treatment effect.
`mean.mu.0`	Prior mean of the mean of the base distribution default value is mean.mu.0 = 0.
`sd.mu.0`	Prior standard deviation of the base distribution, the default value is 10.
`scale.sigma.between`	Prior scale parameter for scale gamma distribution for the precision between studies. The default value is 0.5.
`df.scale.between`	Degrees of freedom of the scale gamma distribution for the precision between studies. The default value is 1, which results in a Half Cauchy distribution for the standard deviation between studies. Larger values e.g. 30 corresponds to a Half Normal distribution.
`alpha.0`	Lower bound of the uniform prior for the concentration parameter for the DPM, default value is alpha.0 = 0.03.
`alpha.1`	Upper bound of the uniform prior for the concentration parameter for the DPM, default value is alpha.1 = 10.
`K`	Maximum number of clusters in the DPM, default value is K = 30.
`nr.chains`	Number of chains for the MCMC computations, default 2.
`nr.iterations`	Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.
`nr.adapt`	Number of iterations in the adaptation process, default is 1000. Some models may need more iterations during adptation.
`nr.burnin`	Number of iteration discard for burn-in period, default is 1000. Some models may need a longer burnin period.
`nr.thin`	Thinning rate, it must be a positive integer, the default value 1.

Details

The results of the object of the class bcmeta can be extracted with R2jags or with rjags. In addition a summary, a print and a plot functions are implemented for this type of object.

Value

This function returns an object of the class "dpmeta". This object contains the MCMC output of each parameter and hyper-parameter in the model and the data frame used for fitting the model.

References

Verde, P.E. (2021) A Bias-Corrected Meta-Analysis Model for Combining Studies of Different Types and Quality. Biometrical Journal; 1–17.

Examples

```
## Not run:
library(jarbes)


# Example: Stemcells

data("stemcells")
stemcells$TE = stemcells$effect.size
stemcells$seTE = stemcells$se.effect

bm1 = dpmmeta(stemcells)
summary(bm1)
plot(bm1, x.lim = c(-1, 7), y.lim = c(0, 1))

diagnostic(bm1, study.names = stemcells$trial,
           post.p.value.cut = 0.05,
           lwd.forest = 0.5, shape.forest = 4)

diagnostic(bm1, post.p.value.cut = 0.05,
           lwd.forest = 0.5, shape.forest = 4)

## End(Not run)
```

`dpmetareg` *Bayesian Meta-Analysis with Dirichlet Process Priors*

Description

This function performers a Bayesian meta-analysis with DP as random effects

Usage

```
dpmetareg(
  data,
  x,
  mean.mu.0 = 0,
  sd.mu.0 = 10,
  scale.sigma.between = 0.5,
  df.scale.between = 1,
  alpha.0 = 0.03,
  alpha.1 = 10,
  K = 30,
  nr.chains = 2,
  nr.iterations = 10000,
  nr.adapt = 1000,
  nr.burnin = 1000,
  nr.thin = 1
)
```

Arguments

data	A data frame with at least two columns with the following names: 1) TE = treatment effect, 2) seTE = the standard error of the treatment effect.
x	a covariate to perform meta-regression.
mean.mu.0	Prior mean of the mean of the base distribution default value is mean.mu.0 = 0.
sd.mu.0	Prior standard deviation of the base distribution, the default value is 10.
scale.sigma.between	Prior scale parameter for scale gamma distribution for the precision between studies. The default value is 0.5.
df.scale.between	Degrees of freedom of the scale gamma distribution for the precision between studies. The default value is 1, which results in a Half Cauchy distribution for the standard deviation between studies. Larger values e.g. 30 corresponds to a Half Normal distribution.
alpha.0	Lower bound of the uniform prior for the concentration parameter for the DPM, default value is alpha.0 = 0.03.
alpha.1	Upper bound of the uniform prior for the concentration parameter for the DPM, default value is alpha.1 = 10.
K	Maximum number of clusters in the DPM, default value is K = 30.
nr.chains	Number of chains for the MCMC computations, default 2.
nr.iterations	Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.
nr.adapt	Number of iterations in the adaptation process, default is 1000. Some models may need more iterations during adptation.
nr.burnin	Number of iteration discard for burn-in period, default is 1000. Some models may need a longer burnin period.
nr.thin	Thinning rate, it must be a positive integer, the default value 1.

Details

The results of the object of the class bcmeta can be extracted with R2jags or with rjags. In addition a summary, a print and a plot functions are implemented for this type of object.

Value

This function returns an object of the class "dpmetareg". This object contains the MCMC output of each parameter and hyper-parameter in the model and the data frame used for fitting the model.

References

Verde, P.E. (2021) A Bias-Corrected Meta-Analysis Model for Combining Studies of Different Types and Quality. Biometrical Journal; 1–17.

Examples

```
## Not run:
library(jarbes)


# Example: Stemcells

data("stemcells")
stemcells$TE = stemcells$effect.size
stemcells$seTE = stemcells$se.effect

bm1 = dpmmeta(stemcells)
summary(bm1)
plot(bm1, x.lim = c(-1, 7), y.lim = c(0, 1))

diagnostic(bm1, study.names = stemcells$trial,
           post.p.value.cut = 0.05,
           lwd.forest = 0.5, shape.forest = 4)

diagnostic(bm1, post.p.value.cut = 0.05,
           lwd.forest = 0.5, shape.forest = 4)

## End(Not run)
```

---

dpmmeta *Bayesian Meta-Analysis with Dirichlet Process Mixtures Priors*

Description

This function performers a Bayesian meta-analysis with DPM as random effects

Usage

```
dpmmeta(
  data,
  mean.mu.0 = 0,
  sd.mu.0 = 10,
  scale.sigma.between = 0.5,
  df.scale.between = 1,
  alpha.0 = 0.03,
  alpha.1 = 10,
  K = 5,
  nr.chains = 2,
  nr.iterations = 10000,
  nr.adapt = 1000,
  nr.burnin = 1000,
  nr.thin = 1
)
```

Arguments

`data`	A data frame with at least two columns with the following names: 1) TE = treatment effect, 2) seTE = the standard error of the treatment effect.
`mean.mu.0`	Prior mean of the mean of the base distribution default value is mean.mu.0 = 0.
`sd.mu.0`	Prior standard deviation of the base distribution, the default value is 10.
`scale.sigma.between`	Prior scale parameter for scale gamma distribution for the precision between studies. The default value is 0.5.
`df.scale.between`	Degrees of freedom of the scale gamma distribution for the precision between studies. The default value is 1, which results in a Half Cauchy distribution for the standard deviation between studies. Larger values e.g. 30 corresponds to a Half Normal distribution.
`alpha.0`	Lower bound of the uniform prior for the concentration parameter for the DPM, default value is alpha.0 = 0.03.
`alpha.1`	Upper bound of the uniform prior for the concentration parameter for the DPM, default value is alpha.1 = 10.
`K`	Maximum number of clusters in the DPM, default value is K = 5.
`nr.chains`	Number of chains for the MCMC computations, default 2.
`nr.iterations`	Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.
`nr.adapt`	Number of iterations in the adaptation process, default is 1000. Some models may need more iterations during adptation.
`nr.burnin`	Number of iteration discard for burn-in period, default is 1000. Some models may need a longer burnin period.
`nr.thin`	Thinning rate, it must be a positive integer, the default value 1.

Details

The results of the object of the class bcmeta can be extracted with R2jags or with rjags. In addition a summary, a print and a plot functions are implemented for this type of object.

Value

This function returns an object of the class "bmeta". This object contains the MCMC output of each parameter and hyper-parameter in the model and the data frame used for fitting the model.

References

Verde, P.E. (2021) A Bias-Corrected Meta-Analysis Model for Combining Studies of Different Types and Quality. Biometrical Journal; 1–17.

Examples

```
## Not run:
library(jarbes)


# Example: Stemcells

data("stemcells")
stemcells$TE = stemcells$effect.size
stemcells$seTE = stemcells$se.effect

bm1 = dpmmeta(stemcells)
summary(bm1)
plot(bm1, x.lim = c(-1, 7), y.lim = c(0, 1))

diagnostic(bm1, study.names = stemcells$trial,
           post.p.value.cut = 0.05,
           lwd.forest = 0.5, shape.forest = 4)

diagnostic(bm1, post.p.value.cut = 0.05,
           lwd.forest = 0.5, shape.forest = 4)

## End(Not run)
```

`effect` *Generic effect function.*

Description

Generic effect function.

Usage

```
effect(object, ...)
```

Arguments

`object`	The object generated by the function hmr.
`...`	...

effect.hmr *Posterior distribution of Effectiveness for a subgroup of patients*

Description

This function estimates the posterior distribution for a subgroup of patients identified with the function hmr (Hierarchical Meta-Regression).

Usage

```
## S3 method for class 'hmr'
effect(
  object,
  B.lower = 0,
  B.upper = 3,
  k = 1,
  level = c(0.5, 0.75, 0.95),
  x.lim = c(-9, 5),
  y.lim = c(-1, 5),
  x.lab = "Baseline risk",
  y.lab = "Effectiveness",
 title.plot = paste("Posterior Effectiveness for a subgroup (50%, 75% and 95%)"),
  kde2d.n = 25,
  marginals = TRUE,
  bin.hist = 30,
  color.line = "black",
  color.hist = "white",
  color.data.points = "black",
  alpha.data.points = 0.1,
  S = 5000,
  display.probability = FALSE,
  line.no.effect = 0,
  font.size.title = 20,
  ...
)
```

Arguments

object	The object generated by the function hmr.
B.lower	Lower limit of bias correction. The default is 0 meaning no bias correction.
B.upper	Upper limit of bias correction. The default is 3 meaning three times bias correction.
k	Covariable number indicating the subgroup.
level	Vector with the probability levels of the contour plot. The default values are: 0.5, 0.75, and 0.95.

x.lim	Numeric vector of length 2 specifying the x-axis limits.
y.lim	Numeric vector of length 2 specifying the y-axis limits.
x.lab	Text with the label of the x-axis.
y.lab	Text with the label of the y-axis.
title.plot	Text for setting a title in the bias plot.
kde2d.n	The number of grid points in each direction for the non-parametric density estimation. The default is 25.
marginals	If TRUE the marginal histograms of the posteriors are added to the plot.
bin.hist	The number of bins in for the histograms. The default value is 30.
color.line	The color of the contour lines. The default is "black.
color.hist	The color of the histogram bars. The default is "white".
color.data.points	The color of the data points. The default is "black".
alpha.data.points	Transparency of the data points.
S	The number of sample values from the joint posterior distribution used to approximate the contours. The default is S=5000.
display.probability	Logical, if TRUE the figure display probabilities.
line.no.effect	Horizontal line used as reference for no effect.
font.size.title	Font size of the title.
...	...

`healing` *Efficacy of diabetic foot healing using adjuvant treatments*

Description

Meta-analysis of 35 randomized controlled trials investigating the effectiveness in the application of adjuvant therapies for diabetic patients compared to medical routine care, where the endpoint was healing without amputations within a period less than or equal to one year.

Format

A matrix with 35 rows and 9 columns. Each row represents study results, the columns are:

Study Name of the first author and year.

n_t Number of patients in the treatment group.

n_c Number of patients in the control group.

y_t Number of heal patients in the treatment group.

y_c Number of heal patients in the control group.

ndrop Total number of drop out patients.

fup_weeks Length of followup in weeks.

PAD Inclusion of patients with peripheral arterial disease.

wagner_4 Inclusion of patients with Wagner score 3 and 4.

Source

The data were obtainded from: Centre for Clinical Practice at NICE (UK and others) (2011), Clinical guideline 119. Diabetic foot problems: Inpatient Management of Diabetic Foot Problems. Tech. rep., National Institute for Health and Clinical Excellence.

References

Verde, P.E. (2018) The Hierarchical Meta-Regression Approach and Learning from Clinical Evidence. Technical Report.

`healingipd` *Individual participant data for diabetic patients*

Description

Prospective cohort study.

Format

A dataframe with 260 rows and 18 columns. Each row represents a patient, the columns are:

healing.without.amp Outcome variable: Healing without amputation with in one year.

duration_lesion_days Duration of leasions in days at baseline.

PAD Peripheral arterial disease yes/no.

neuropathy Neuropathy yes/no.

first.ever.lesion First ever lesion yes/no.

no.continuous.care No continuous care yes/no.

male yes/no.

diab.typ2 Diabetes type 2 yes/no.

insulin Insulin dependent yes/no.

HOCHD HOCHD yes/no.

HOS HOCHD yes/no.

CRF CRF yes/no.

dialysis Dialysis yes/no.

DNOAP DNOAP yes/no.

smoking.ever Ever smoke yes/no.

age Age at baseline in years.

diabdur Diabetes duration at baseline.

wagner.class Wagner score 1-2 vs. 3-4-5.

Source

Morbach, S, et al. (2012). Long-Term Prognosis of Diabetic Foot Patients and Their Limbs: Amputation and death over the course of a decade,Diabetes Care, 35, 10, 2012-2017.

References

Verde, P.E. (2018) The Hierarchical Meta-Regression Approach and Learning from Clinical Evidence. Technical Report.

`hips` *Meta-analysis:*

Description

Meta-analysis of 15 studies investigating total hip replacement to compare the risk of revision of cemented and uncemented implantfixation modalities, by pooling treatment effectestimates from OS and RCTs.

Format

A dataframe with 15 rows and 12 columns. Each row represents study results, the columns are:

Study Author and year.

Study_type Study desing.

N_of_revisions Number of revisions.

Total_cemented Total number of cemmented cases.

N_of_revisions_uncemented Number of uncemented revisions.

Total_uncemented Total number of uncemmented cases.

Relative_risks_computed RR calculated from the two by two table.

L95CI Lower 95prc CI

U95CI Upper 95prc CI

mean_age Mean age of the study

proportion_of_women Proportion of women in the study.

Follow_up Time to follow-up in years.

Source

Schnell-Inderst P, Iglesias CP, Arvandi M, Ciani O, Matteucci Gothe R, Peters J, Blom AW, Taylor RS and Siebert U (2017). A bias-adjusted evidence synthesis of RCT and observational data: the case of total hip replacement. Health Econ. 26(Suppl. 1): 46–69.

hmr *Bayesian meta-analysis to combine aggregated and individual participant data for cross design synthesis.*

Description

This function performers a Bayesian cross design synthesis. The function fits a hierarchical meta-regression model based on a bivariate random effects model.

Usage

```
hmr(
  data,
  two.by.two = TRUE,
  dataIPD,
  re = "normal",
  link = "logit",
  mean.mu.1 = 0,
  mean.mu.2 = 0,
  mean.mu.phi = 0,
  sd.mu.1 = 1,
  sd.mu.2 = 1,
  sd.mu.phi = 1,
  sigma.1.upper = 5,
  sigma.2.upper = 5,
  sigma.beta.upper = 5,
  mean.Fisher.rho = 0,
  sd.Fisher.rho = 1/sqrt(2),
  df = 4,
  df.estimate = FALSE,
  df.lower = 3,
  df.upper = 20,
  split.w = FALSE,
  nr.chains = 2,
  nr.iterations = 10000,
  nr.adapt = 1000,
  nr.burnin = 1000,
  nr.thin = 1
)
```

Arguments

data	Aggregated data results: a data frame where the first four columns containing the number of events in the control group (yc), the number of patients in the control group (nc), the number of events in the treatment group (yt) and the number of patients in the treatment group (nt). If two.by.two = TRUE a data frame where each line contains the trial results with column names: yc, nc, yt, nt.

two.by.two	If TRUE indicates that the trial results are with names: yc, nc, yt, nt.
dataIPD	Individual participant data: a data frame where the first column is the outcome variable and the other columns represent individual participant charachteristics.
re	Random effects distribution for the resulting model. Possible values are *normal* for bivariate random effects and *sm* for scale mixtures.
link	The link function used in the model. Possible values are *logit*, *cloglog probit*.
mean.mu.1	Prior mean of baseline risk, default value is 0.
mean.mu.2	Prior mean of treatment effect, default value is 0.
mean.mu.phi	Prior mean of the bias parameter which measures the difference between the baseline mean mu.1 and the intercept parameter of the logistic regression of the individual participant data. The defalut vaule is 0.
sd.mu.1	Prior standard deviation of mu.1, default value is 1. The default prior of mu.1 is a logistic distribution with mean 0 and dispersion 1. The implicit prior for mu.1 in the probability scale is a uniform between 0 and 1.
sd.mu.2	Prior standard deviation of mu.2, default value is 1. The default prior of mu.2 is a logistic distribution with mean 0 and dispersion 1. The implicit prior for mu.2 in the probability scale is a uniform between 0 and 1.
sd.mu.phi	Prior standard deviation of mu.phi, default value is 1.
sigma.1.upper	Upper bound of the uniform prior of sigma.1, default value is 5.
sigma.2.upper	Upper bound of the uniform prior of sigma.2, default value is 5.
sigma.beta.upper	Upper bound of the uniform prior of sigma.beta, default value is 5.
mean.Fisher.rho	Mean of rho in the Fisher scale, default value is 0.
sd.Fisher.rho	Standard deviation of rho in the Fisher scale, default value is 1/sqrt(2).
df	If de.estimate = FALSE, then df is the degrees of freedom for the scale mixture distribution, default value is 4.
df.estimate	Estimate the posterior of df. The default value is FALSE.
df.lower	Lower bound of the prior of df. The default value is 3.
df.upper	Upper bound of the prior of df. The default value is 30.
split.w	Split the w parameter in two independent weights one for each random effect. The default value is FALSE.
nr.chains	Number of chains for the MCMC computations, default 5.
nr.iterations	Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.
nr.adapt	Number of iterations in the adaptation process, default is 1000. Some models may need more iterations during adptation.
nr.burnin	Number of iteration discarded for burnin period, default is 1000. Some models may need a longer burnin period.
nr.thin	Thinning rate, it must be a positive integer, the default value 1.

Details

The number of events in the control and treated group are modeled with two conditional Binomial distributions and the random-effects are based on a bivariate scale mixture of Normals.

The individual participant data is modeled as a Bayesian logistic regression for participants in the control group. Coefficients in the regression are modeled as exchangeables.

The function calculates the implicit hierarchical meta-regression, where the treatment effect is regressed to the baseline risk (rate of events in the control group). The scale mixture weights are used to adjust for internal validity and structural outliers identification.

The implicit hierarchical meta-regression is used to predict the treatment effect for subgroups of individual participant data.

Computations are done by calling JAGS (Just Another Gibbs Sampler) to perform MCMC (Markov Chain Monte Carlo) sampling and returning an object of the class *mcmc.list*.

Installation of JAGS: It is important to note that R 3.3.0 introduced a major change in the use of toolchain for Windows. This new toolchain is incompatible with older packages written in C++. As a consequence, if the installed version of JAGS does not match the R installation, then the rjags package will spontaneously crash. Therefore, if a user works with R version >= 3.3.0, then JAGS must be installed with the installation program JAGS-4.2.0-Rtools33.exe. For users who continue using R 3.2.4 or an earlier version, the installation program for JAGS is the default installer JAGS-4.2.0.exe.

Value

This function returns an object of the class "hmr". This object contains the MCMC output of each parameter and hyper-parameter in the model, the data frame used for fitting the model, the link function, type of random effects distribution and the splitting information for conflict of evidence analysis.

The results of the object of the class hmr can be extracted with R2jags. In addition a summary, a print and a plot function are implemented for this type of object.

References

Verde, P.E, Ohmann, C., Icks, A. and Morbach, S. (2016) Bayesian evidence synthesis and combining randomized and nonrandomized results: a case study in diabetes. Statistics in Medicine. Volume 35, Issue 10, 10 May 2016, Pages: 1654 to 1675.

Verde, P.E. (2017) The hierarchical meta-regression approach and learning from clinical evidence. Submited to the Biometrical Journal.

Verde, P.E. (2018) The Hierarchical Meta-Regression Approach and Learning from Clinical Evidence. Technical report.

Examples

```
## Not run:
library(jarbes)

data("healing")
AD <- healing[, c("y_c", "n_c", "y_t", "n_t")]
```

```
data("healingipd")

IPD <- healingipd[, c("healing.without.amp", "PAD", "neuropathy",
"first.ever.lesion", "no.continuous.care", "male", "diab.typ2",
"insulin", "HOCHD", "HOS", "CRF", "dialysis", "DNOAP", "smoking.ever",
"diabdur", "wagner.class")]

mx2 <- hmr(AD, two.by.two = FALSE,
           dataIPD = IPD,
           re = "sm",
           link = "logit",
           sd.mu.1 = 2,
           sd.mu.2 = 2,
           sd.mu.phi = 2,
           sigma.1.upper = 5,
           sigma.2.upper = 5,
           sigma.beta.upper = 5,
           sd.Fisher.rho = 1.25,
           df.estimate = FALSE,
           df.lower = 3,
           df.upper = 10,
           nr.chains = 1,
           nr.iterations = 1500,
           nr.adapt = 100,
           nr.thin = 1)

print(mx2)



# This experiment corresponds to Section 4 in Verde (2018).
#
# Experiment: Combining aggretated data from RCTs and a single
# observational study with individual participant data.
#
# In this experiment we assess conflict of evidence between the RCTs
# and the observational study with a partially identified parameter
# mu.phi.
#
# We run two simulated data: 1) mu.phi = 0.5 which is diffucult to
# identify. 2) mu.phi = 2 which can be identify. The simulations are
# used to see if the hmr() function can recover mu.phi.
#


library(MASS)
library(ggplot2)
library(jarbes)
library(gridExtra)
library(mcmcplots)
```

```
# Simulation of the IPD data

invlogit <- function (x)
{
 1/(1 + exp(-x))
}

# Data set for mu.phi = 0.5 ....................................

# Parameters values
mu.phi.true <- 0.5
beta0 <- mu.1.true + mu.phi.true
beta1 <- 2.5
beta2 <- 2

# Regression variables

x1 <- rnorm(200)
x2 <- rbinom(200, 1, 0.5)

# Binary outcome as a function of "b0 + b1 * x1 + b2 * x2"

y <- rbinom(200, 1,
         invlogit(beta0 + beta1 * x1 + beta2 * x2))


# Preparing the plot to visualize the data
jitter.binary <- function(a, jitt = 0.05)

 ifelse(a==0, runif(length(a), 0, jitt),
        runif(length(a), 1-jitt, 1))



plot(x1, jitter.binary(y), xlab = "x1",
    ylab = "Success probability")

curve(invlogit(beta0 + beta1*x),
     from = -2.5, to = 2.5, add = TRUE, col ="blue", lwd = 2)
curve(invlogit(beta0 + beta1*x + beta2),
     from = -2.5, to = 2.5, add = TRUE, col ="red", lwd =2)
legend("bottomright", c("b2 = 0", "b2 = 2"),
      col = c("blue", "red"), lwd = 2, lty = 1)

noise <- rnorm(100*20)
dim(noise) <- c(100, 20)
n.names <- paste(rep("x", 20), seq(3, 22), sep="")
colnames(noise) <- n.names

data.IPD <- data.frame(y, x1, x2, noise)
```

```
# Application of HMR ......................................

res.s2 <- hmr(AD.s1, two.by.two = FALSE,
              dataIPD = data.IPD,
              sd.mu.1 = 2,
              sd.mu.2 = 2,
              sd.mu.phi = 2,
              sigma.1.upper = 5,
              sigma.2.upper = 5,
              sd.Fisher.rho = 1.5)


print(res.s2)

# Data set for mu.phi = 2 ................................
# Parameters values

mu.phi.true <- 2
beta0 <- mu.1.true + mu.phi.true
beta1 <- 2.5
beta2 <- 2

# Regression variables
x1 <- rnorm(200)
x2 <- rbinom(200, 1, 0.5)
# Binary outcome as a function of "b0 + b1 * x1 + b2 * x2"
y <- rbinom(200, 1,
            invlogit(beta0 + beta1 * x1 + beta2 * x2))

# Preparing the plot to visualize the data
jitter.binary <- function(a, jitt = 0.05)

 ifelse(a==0, runif(length(a), 0, jitt),
        runif(length(a), 1-jitt, 1))

plot(x1, jitter.binary(y), xlab = "x1",
    ylab = "Success probability")

curve(invlogit(beta0 + beta1*x),
     from = -2.5, to = 2.5, add = TRUE, col ="blue", lwd = 2)
curve(invlogit(beta0 + beta1*x + beta2),
     from = -2.5, to = 2.5, add = TRUE, col ="red", lwd =2)
legend("bottomright", c("b2 = 0", "b2 = 2"),
      col = c("blue", "red"), lwd = 2, lty = 1)

noise <- rnorm(100*20)
dim(noise) <- c(100, 20)
n.names <- paste(rep("x", 20), seq(3, 22), sep="")
colnames(noise) <- n.names

data.IPD <- data.frame(y, x1, x2, noise)

# Application of HMR ......................................
```

```
res.s3 <- hmr(AD.s1, two.by.two = FALSE,
              dataIPD = data.IPD,
              sd.mu.1 = 2,
              sd.mu.2 = 2,
              sd.mu.phi = 2,
              sigma.1.upper = 5,
              sigma.2.upper = 5,
              sd.Fisher.rho = 1.5
)

print(res.s3)

# Posteriors for mu.phi ..........................
attach.jags(res.s2)
mu.phi.0.5 <- mu.phi
df.phi.05 <- data.frame(x = mu.phi.0.5)

attach.jags(res.s3)
mu.phi.1 <- mu.phi
df.phi.1 <- data.frame(x = mu.phi.1)


p1 <- ggplot(df.phi.05, aes(x=x))+
 xlab(expression(mu[phi])) +
 ylab("Posterior distribution")+
 xlim(c(-7,7))+
 geom_histogram(aes(y=..density..),fill = "royalblue",
              colour = "black", alpha= 0.4, bins=60) +
 geom_vline(xintercept = 0.64, colour = "black", size = 1.7, lty = 2)+
 geom_vline(xintercept = 0.5, colour = "black", size = 1.7, lty = 1)+
 stat_function(fun = dlogis,
               n = 101,
               args = list(location = 0, scale = 1), size = 1.5) + theme_bw()

p2 <- ggplot(df.phi.1, aes(x=x))+
 xlab(expression(mu[phi])) +
 ylab("Posterior distribution")+
 xlim(c(-7,7))+
 geom_histogram(aes(y=..density..),fill = "royalblue",
                colour = "black", alpha= 0.4, bins=60) +
 geom_vline(xintercept = 2.2, colour = "black", size = 1.7, lty = 2)+
 geom_vline(xintercept = 2, colour = "black", size = 1.7, lty = 1)+
 stat_function(fun = dlogis,
               n = 101,
               args = list(location = 0, scale = 1), size = 1.5) + theme_bw()

grid.arrange(p1, p2, ncol = 2, nrow = 1)


# Cater plots for regression coefficients ........................
```

```
var.names <- names(data.IPD[-1])
v <- paste("beta", names(data.IPD[-1]), sep = ".")
mcmc.x <- as.rjags.mcmc(res.s2$BUGSoutput$sims.matrix)
mcmc.x.2 <- as.mcmc.rjags(res.s2)
mcmc.x.3 <- as.mcmc.rjags(res.s3)

greek.names <- paste(paste("beta[",1:22, sep=""),"]", sep="")
par.names <- paste(paste("beta.IPD[",1:22, sep=""),"]", sep="")

caterplot(mcmc.x.2,
         parms = par.names,
         col = "black", lty = 1,
         labels = greek.names,
         greek = T,
         labels.loc="axis", cex =0.7,
         style = "plain",reorder = F, denstrip = F)

caterplot(mcmc.x.3,
         parms = par.names,
         col = "grey", lty = 2,
         labels = greek.names,
         greek = T,
         labels.loc="axis", cex =0.7,
         style = "plain",reorder = F, denstrip = F,
         add = TRUE,
         collapse=TRUE, cat.shift=-0.5)



abline(v=0, lty = 2, lwd = 2)
abline(v =2, lty = 2, lwd = 2)
abline(v =2.5, lty = 2, lwd = 2)

# End of the examples.


## End(Not run)
```

`metarisk` *Bayesian meta-analysis for using baseline risk adjustment*

Description

This function performers a Bayesian meta-analysis to analyse heterogeneity of the treatment effect as a function of the baseline risk. The function fits a hierarchical meta-regression model based on a bivariate random effects model.

Usage

```
metarisk(
  data,
  two.by.two = TRUE,
  re = "normal",
  link = "logit",
  mean.mu.1 = 0,
  mean.mu.2 = 0,
  sd.mu.1 = 1,
  sd.mu.2 = 1,
  sigma.1.upper = 5,
  sigma.2.upper = 5,
  mean.Fisher.rho = 0,
  sd.Fisher.rho = 1/sqrt(2),
  df = 4,
  df.estimate = FALSE,
  df.lower = 3,
  df.upper = 20,
  split.w = FALSE,
  nr.chains = 2,
  nr.iterations = 10000,
  nr.adapt = 1000,
  nr.burnin = 1000,
  nr.thin = 1
)
```

Arguments

`data`	A data frame where the first four columns containing the number of events in the control group (yc), the number of patients in the control group (nc), the number of events in the treatment group (yt) and the number of patients in the treatment group (nt). If two.by.two = TRUE a data frame where each line contains the trial results with column names: yc, nc, yt, nt.
`two.by.two`	If TRUE indicates that the trial results are with names: yc, nc, yt, nt.
`re`	Random effects distribution for the resulting model. Possible values are *normal* for bivariate random effects and *sm* for scale mixtures.
`link`	The link function used in the model. Possible values are *logit*, *cloglog probit*.
`mean.mu.1`	Prior mean of baseline risk, default value is 0.
`mean.mu.2`	Prior mean of the relative treatment effect, default value is 0.
`sd.mu.1`	Prior standard deviation of mu.1, default value is 1. The default prior of mu.1 is a logistic distribution with mean 0 and dispersion 1. The implicit prior for mu.1 in the probability scale is a uniform between 0 and 1.
`sd.mu.2`	Prior standard deviation of mu.2, default value is 1. The default prior of mu.2 is a logistic distribution with mean 0 and dispersion 1. The implicit prior for mu.2 in the probability scale is a uniform between 0 and 1.
`sigma.1.upper`	Upper bound of the uniform prior of sigma.1, default value is 5.

`sigma.2.upper`	Upper bound of the uniform prior of sigma.2, default value is 5.
`mean.Fisher.rho`	Mean of rho in the Fisher scale default value is 0.
`sd.Fisher.rho`	Standard deviation of rho in the Fisher scale, default value is 1/sqrt(2).
`df`	If de.estimate = FALSE, then df is the degrees of freedom for the scale mixture distribution, default value is 4.
`df.estimate`	Estimate the posterior of df. The default value is FALSE.
`df.lower`	Lower bound of the prior of df. The default value is 3.
`df.upper`	Upper bound of the prior of df. The default value is 30.
`split.w`	Split the w parameter in two independent weights one for each random effect. The default value is FALSE.
`nr.chains`	Number of chains for the MCMC computations, default 5.
`nr.iterations`	Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.
`nr.adapt`	Number of iterations in the adaptation process, defualt is 1000. Some models may need more iterations during adptation.
`nr.burnin`	Number of iteration discared for burnin period, default is 1000. Some models may need a longer burnin period.
`nr.thin`	Thinning rate, it must be a positive integer, the default value is 1.

Details

The number of events in the control and treated group are modeled with two conditional Binomial distributions and the random-effects are based on a bivariate scale mixture of Normals.

The function calculates the implicit hierarchical meta-regression, where the treatment effect is regressed to the baseline risk (rate of events in the control group). The scale mixture weights are used to adjust for internal validity and structural outliers identification.

Computations are done by calling JAGS (Just Another Gibbs Sampler) to perform MCMC (Markov Chain Monte Carlo) sampling and returning an object of the class *mcmc.list*.

Value

This function returns an object of the class "metarisk". This object contains the MCMC output of each parameter and hyper-parameter in the model, the data frame used for fitting the model, the link function, type of random effects distribution and the splitting information for conflict of evidence analysis.

The results of the object of the class metadiag can be extracted with R2jags or with rjags. In addition a summary, a print and a plot functions are implemented for this type of object.

References

Verde, P.E. and Curcio, D. (2019) Hierarchical Meta-Regression Modelling: The Case of The Pneumococcal Polysaccharide Vaccine. Technical Report.

Verde, P.E. (2019) The hierarchical meta-regression approach and learning from clinical evidence. Biometrical Journal. 1 - 23.

Verde, P. E. (2017) Two Examples of Bayesian Evidence Synthesis with the Hierarchical Meta-Regression Approach. Chap.9, pag 189-206. Bayesian Inference, ed. Tejedor, Javier Prieto. InTech.

Examples

```
## Not run:
library(jarbes)

# This example is used to test the function and it runs in about 5 seconds.
# In a real application you must increase the number of MCMC interations.
# For example use: nr.burnin = 5000 and nr.iterations = 20000




# The following examples corresponds to Section 4 in Verde (2019).
# These are simulated examples to investigate internal and
# external validity bias in meta-analysis.


library(MASS)
library(ggplot2)
library(jarbes)
library(gridExtra)
library(mcmcplots)

#Experiment 1: External validity bias

set.seed(2018)
# mean control
pc <- 0.7
# mean treatment
pt <- 0.4
# reduction of "amputations" odds ratio
OR <- (pt/(1-pt)) /(pc/(1-pc))
OR

# mu_2
log(OR)
mu.2.true <- log(OR)
#sigma_2
sigma.2.true <- 0.5
# mu_1
mu.1.true <- log(pc/(1-pc))
mu.1.true
#sigma_1
sigma.1.true <- 1
# rho
rho.true <- -0.5
Sigma <- matrix(c(sigma.1.true^2, sigma.1.true*sigma.2.true*rho.true,
                  sigma.1.true*sigma.2.true*rho.true, sigma.2.true^2),
                  byrow = TRUE, ncol = 2)
```

```
Sigma

theta <- mvrnorm(35, mu = c(mu.1.true, mu.2.true),
                 Sigma = Sigma )


plot(theta, xlim = c(-2,3))
abline(v=mu.1.true, lty = 2)
abline(h=mu.2.true, lty = 2)
abline(a = mu.2.true, b=sigma.2.true/sigma.1.true * rho.true, col = "red")
abline(lm(theta[,2]~theta[,1]), col = "blue")

# Target group
mu.T <- mu.1.true + 2 * sigma.1.true
abline(v=mu.T, lwd = 3, col = "blue")
eta.true <- mu.2.true + sigma.2.true/sigma.1.true*rho.true* mu.T
eta.true
exp(eta.true)
abline(h = eta.true, lty =3, col = "blue")
# Simulation of each primary study:
n.c <- round(runif(35, min = 30, max = 60),0)
n.t <- round(runif(35, min = 30, max = 60),0)
y.c <- y.t <- rep(0, 35)
p.c <- exp(theta[,1])/(1+exp(theta[,1]))
p.t <- exp(theta[,2]+theta[,1])/(1+exp(theta[,2]+theta[,1]))
for(i in 1:35)
{
 y.c[i] <- rbinom(1, n.c[i], prob = p.c[i])
 y.t[i] <- rbinom(1, n.t[i], prob = p.t[i])
}

AD.s1 <- data.frame(yc=y.c, nc=n.c, yt=y.t, nt=n.t)

############################################################
incr.e <- 0.05
incr.c <- 0.05
n11 <- AD.s1$yt
n12 <- AD.s1$yc
n21 <- AD.s1$nt - AD.s1$yt
n22 <- AD.s1$nc - AD.s1$yc
AD.s1$TE <- log(((n11 + incr.e)*(n22 + incr.c))/((n12 + incr.e) * (n21 + incr.c)))
AD.s1$seTE <- sqrt((1/(n11 + incr.e) + 1/(n12 + incr.e) +
                     1/(n21 + incr.c) + 1/(n22 + incr.c)))

Pc <- ((n12 + incr.c)/(AD.s1$nc + 2*incr.c))

AD.s1$logitPc <- log(Pc/(1-Pc))

AD.s1$Ntotal <- AD.s1$nc + AD.s1$nt
rm(list=c("Pc", "n11","n12","n21","n22","incr.c", "incr.e"))


dat.points <- data.frame(TE = AD.s1$TE, logitPc = AD.s1$logitPc, N.total = AD.s1$Ntotal)
```

```
#####################################################################

res.s1 <- metarisk(AD.s1, two.by.two = FALSE, sigma.1.upper = 5,
                   sigma.2.upper = 5,
                   sd.Fisher.rho = 1.5)

print(res.s1, digits = 4)


library(R2jags)
attach.jags(res.s1)
eta.hat <- mu.2 + rho*sigma.2/sigma.1*(mu.T - mu.1)
mean(eta.hat)
sd(eta.hat)

OR.eta.hat <- exp(eta.hat)
mean(OR.eta.hat)
sd(OR.eta.hat)
quantile(OR.eta.hat, probs = c(0.025, 0.5, 0.975))

ind.random <- sample(1:18000, size = 80, replace = FALSE)

#############################################################
p1 <- ggplot(dat.points, aes(x = logitPc, y = TE, size = N.total)) +
      xlab("logit Baseline Risk")+
      ylab("log(Odds Ratio)")+
      geom_point(shape = 21, colour = "blue") + scale_size_area(max_size=10)+
      scale_x_continuous(name= "Rate of The Control Group (logit scale)",
                         limits=c(-2, 5)) +
     geom_vline(xintercept = mu.T, colour = "blue", size = 1, lty = 1) +
       geom_hline(yintercept = eta.true, colour = "blue", size = 1, lty = 1)+
         geom_abline(intercept=beta.0[ind.random],
                     slope=beta.1[ind.random],alpha=0.3,
                     colour = "grey", size = 1.3, lty = 2)+
         geom_abline(intercept = mean(beta.0[ind.random]),
         slope = mean(beta.1[ind.random]),
         colour = "black", size = 1.3, lty = 2)+
     geom_abline(intercept = mu.2.true, slope = sigma.2.true/sigma.1.true * rho.true,
     colour = "blue", size = 1.2)+ theme_bw()


# Posterior of eta.hat

eta.df <- data.frame(x = OR.eta.hat)


p2 <- ggplot(eta.df, aes(x = x)) +
 xlab("Odds Ratio") +
 ylab("Posterior distribution")+
 geom_histogram(fill = "royalblue", colour = "black", alpha= 0.4, bins=80) +
 geom_vline(xintercept = exp(eta.true), colour = "black", size = 1.7, lty = 1)+
 geom_vline(xintercept = c(0.28, 0.22, 0.35), colour = "black", size = 1, lty = 2)+
 theme_bw()
```

```
grid.arrange(p1, p2, ncol = 2, nrow = 1)

#Experiment 2: Internal validity bias and assesing conflict of evidence between the RCTs.


set.seed(2018)
ind <- sample(1:35, size=5, replace = FALSE)
ind
AD.s4.contaminated <- AD.s1[ind,1:4]
AD.s4.contaminated$yc <- AD.s1$yt[ind]
AD.s4.contaminated$yt <- AD.s1$yc[ind]
AD.s4.contaminated$nc <- AD.s1$nt[ind]
AD.s4.contaminated$nt <- AD.s1$nc[ind]
AD.s4.contaminated <- rbind(AD.s4.contaminated,
                         AD.s1[-ind,1:4])

res.s4 <- metarisk(AD.s4.contaminated,
                   two.by.two = FALSE,
                   re = "sm",
                   sigma.1.upper = 3,
                   sigma.2.upper = 3,
                   sd.Fisher.rho = 1.5,
                   df.estimate = TRUE)

print(res.s4, digits = 4)

attach.jags(res.s4)

w.s <- apply(w, 2, median)
w.u <- apply(w, 2, quantile, prob = 0.75)
w.l <- apply(w, 2, quantile, prob = 0.25)

studies <- c(ind,c(1,3,4,5,6,8,9,10,11,13,14,17,18,19,20:35))


dat.weights <- data.frame(x = studies,
                          y = w.s,
                         ylo  = w.l,
                         yhi  = w.u)

# Outliers:
w.col <- studies %in% ind
w.col.plot <- ifelse(w.col, "black", "grey")
w.col.plot[c(9,17, 19,27,34,35)] <- "black"

w.plot <- function(d){
  # d is a data frame with 4 columns
  # d$x gives variable names
  # d$y gives center point
  # d$ylo gives lower limits
  # d$yhi gives upper limits
```

```
  p <- ggplot(d, aes(x=x, y=y, ymin=ylo, ymax=yhi) )+
       geom_pointrange( colour=w.col.plot, lwd =0.8)+
       coord_flip() + geom_hline(yintercept = 1, lty=2)+
       xlab("Study ID") +
       ylab("Scale mixture weights") + theme_bw()
       return(p)}

w.plot(dat.weights)

#List of other possible statistical models:
#    1) Different link functions: logit, cloglog and probit

#    2) Different random effects distributions:
#       "normal" or "sm = scale mixtures".

#    3) For the scale mixture random effects:
#       split.w = TRUE => "split the weights".

#    4) For the scale mixture random effects:
#       df.estimate = TRUE => "estimate the degrees of freedom".

#    5) For the scale mixture random effects:
#       df.estimate = TRUE => "estimate the degrees of freedom".

#    6) For the scale mixture random effects:
#       df = 4 => "fix the degrees of freedom to a particual value".
#       Note that df = 1 fits a Cauchy bivariate distribution to
#       the random effects.
#End of the examples


## End(Not run)
```

`plot.b3lmeta` *Generic plot function for b3lmeta object in jarbes.*

Description

Generic plot function for b3lmeta object in jarbes.

Generic plot function for b3lmeta object in jarbes.

Usage

```
## S3 method for class 'b3lmeta'
plot(
  x,
```

```
  x.lim = c(-3, 3),
  y.lim = c(0, 2.7),
  x.lab = "Treatment Effect: log(OR)",
  y.lab = "Posterior",
  title.plot.1 = "Mean Design Components",
  title.plot.2 = "Three Levels Bayesian Meta-Analysis",
  ...
)

## S3 method for class 'b3lmeta'
plot(
  x,
  x.lim = c(-3, 3),
  y.lim = c(0, 2.7),
  x.lab = "Treatment Effect: log(OR)",
  y.lab = "Posterior",
  title.plot.1 = "Mean Design Components",
  title.plot.2 = "Three Levels Bayesian Meta-Analysis",
  ...
)
```

Arguments

x	The object generated by the b3lmeta function.
x.lim	Numeric vector of length 2 specifying the x-axis limits.
y.lim	Numeric vector of length 2 specifying the y-axis limits.
x.lab	Text with the label of the x-axis.
y.lab	Text with the label of the y-axis.
title.plot.1	Text for the posterior means by design.
title.plot.2	Text for the posterior pooled mean.
...	...

---

plot.bcdpmeta *Generic plot function for bcdpmeta object in jarbes.*

Description

Generic plot function for bcdpmeta object in jarbes.

Usage

```
## S3 method for class 'bcdpmeta'
plot(
  x,
  x.lim = c(-3, 3),
```

```
  y.lim = c(0, 2),
  x.lab = "Treatment Effect: log(OR)",
  y.lab = "Posterior",
  title.plot.1 = "Model Components",
  title.plot.2 = "Bias Corrected Meta-Analysis",
  ...
)
```

Arguments

x	The object generated by the bcmeta function.
x.lim	Numeric vector of length 2 specifying the x-axis limits.
y.lim	Numeric vector of length 2 specifying the y-axis limits.
x.lab	Text with the label of the x-axis.
y.lab	Text with the label of the y-axis.
title.plot.1	Text for the posterior means by component (biased and bias corrected).
title.plot.2	Text for the posterior mean (pooled and predictive).
...	...

---

plot.bcmeta *Generic plot function for bcmeta object in jarbes.*

Description

Generic plot function for bcmeta object in jarbes.

Usage

```
## S3 method for class 'bcmeta'
plot(
  x,
  x.lim = c(-3, 3),
  y.lim = c(0, 2),
  x.lab = "Treatment Effect: log(OR)",
  y.lab = "Posterior",
  title.plot.1 = "Model Components",
  title.plot.2 = "Bias Corrected Meta-Analysis",
  ...
)
```

Arguments

x	The object generated by the bcmeta function.
x.lim	Numeric vector of length 2 specifying the x-axis limits.
y.lim	Numeric vector of length 2 specifying the y-axis limits.
x.lab	Text with the label of the x-axis.
y.lab	Text with the label of the y-axis.
title.plot.1	Text for the posterior means by component (biased and bias corrected).
title.plot.2	Text for the posterior mean (pooled and predictive).
...	...

---

plot.bmeta *Generic plot function for bmeta object in jarbes.*

Description

Generic plot function for bmeta object in jarbes.

Usage

```
## S3 method for class 'bmeta'
plot(
  x,
  x.lim = c(-3, 3),
  y.lim = c(0, 2),
  x.lab = "Treatment Effect: log(OR)",
  y.lab = "Posterior",
  title.plot = "Simple Bayesian Meta-Analysis",
  ...
)
```

Arguments

x	The object generated by the bmeta function.
x.lim	Numeric vector of length 2 specifying the x-axis limits.
y.lim	Numeric vector of length 2 specifying the y-axis limits.
x.lab	Text with the label of the x-axis.
y.lab	Text with the label of the y-axis.
title.plot	Text for setting a title in the plot.
...	...

plot.dpmeta *Generic plot function for bmeta object in jarbes.*

Description

Generic plot function for bmeta object in jarbes.

Usage

```
## S3 method for class 'dpmeta'
plot(
  x,
  x.lim = c(-3, 3),
  y.lim = c(0, 2),
  x.lab = "Treatment Effect: log(OR)",
  y.lab = "Posterior",
  title.plot = "Simple Bayesian Meta-Analysis",
  ...
)
```

Arguments

x	The object generated by the bmeta function.
x.lim	Numeric vector of length 2 specifying the x-axis limits.
y.lim	Numeric vector of length 2 specifying the y-axis limits.
x.lab	Text with the label of the x-axis.
y.lab	Text with the label of the y-axis.
title.plot	Text for setting a title in the plot.
...	...

plot.dpmmeta *Generic plot function for dpmeta object in jarbes.*

Description

Generic plot function for dpmeta object in jarbes.

Usage

```
## S3 method for class 'dpmmeta'
plot(
  x,
  x.lim = c(-3, 3),
  y.lim = c(0, 2),
  x.lab = "Treatment Effect: log(OR)",
  y.lab = "Posterior",
  title.plot = "Simple Bayesian Meta-Analysis",
  ...
)
```

Arguments

x	The object generated by the bmeta function.
x.lim	Numeric vector of length 2 specifying the x-axis limits.
y.lim	Numeric vector of length 2 specifying the y-axis limits.
x.lab	Text with the label of the x-axis.
y.lab	Text with the label of the y-axis.
title.plot	Text for setting a title in the plot.
...	...

---

plot.hmr *Generic plot function for hmr object in jarbes.*

Description

Generic plot function for hmr object in jarbes.

Usage

```
## S3 method for class 'hmr'
plot(
  x,
  x.lim = c(-5, 2.8),
  y.lim = c(-2, 1),
  x.lab = "Event rate of The Control Group (logit scale)",
  y.lab = "No improvement <- Effectiveness -> Improvement",
  title.plot = "HMR: Effectiveness Against Baseline Risk",
  AD.colour = "red",
  IPD.colour = "blue",
  Study.Types = c("AD-RCTs", "IPD-RWD"),
  ...
)
```

Arguments

x	The object generated by the hmr function.
x.lim	Numeric vector of length 2 specifying the x-axis limits.
y.lim	Numeric vector of length 2 specifying the y-axis limits.
x.lab	Text with the label of the x-axis.
y.lab	Text with the label of the y-axis.
title.plot	Text for setting a title in the plot.
AD.colour	Colour of the location of the baseline risk of the aggregated data AD
IPD.colour	Colour of the location of the baseline risk of the individual participant data (IPD) data
Study.Types	Vector of text for the label of the study types
...	...

---

plot.metarisk *Generic plot function for metarisk object in jarbes.*

Description

Generic plot function for metarisk object in jarbes.

Usage

```
## S3 method for class 'metarisk'
plot(
  x,
  x.lim = c(-5, 2.8),
  y.lim = c(-2, 1),
  x.lab = "Rate of The Control Group (logit scale)",
  y.lab = "No improvement <- Treatment effect -> Improvement",
  title.plot = "Treatment Effect Against Baseline Risk",
  ...
)
```

Arguments

x	The object generated by the metarisk function.
x.lim	Numeric vector of length 2 specifying the x-axis limits.
y.lim	Numeric vector of length 2 specifying the y-axis limits.
x.lab	Text with the label of the x-axis.
y.lab	Text with the label of the y-axis.
title.plot	Text for setting a title in the plot.
...	...

ppvcap *Efficacy of Pneumococcal Polysaccharide Vaccine in Preventing Community Acquired Pneumonia*

Description

PPV23 (23-valent pneumococcal polysaccharide vaccine) with 16 Randomized Clinical Trials (RCTs); outcome variable CAP (community-acquired pneumonia).

This data frame corresponds to 16 randomized control trials (RCTs) reporting efficacy of the PPV (Pneumococcal Polysaccharide) vaccine in preventing CAP (community acquired pneumonia). The data frame contains the evaluation of Risk of Bias (RoB) of the trials and some study population characteristics.

Format

A matrix with 16 rows and 18 columns. Each row represents study results, the columns are:

Name_Year Name of the first author and year.

Year Year of publication.

yt Number of infections in the intervention group.

nt Number of patients in the intervention group.

yc Number of infections in the control group.

nc Number of patients in the control group.

TE Treatment Effect as Log Odds Ratio.

seTE Standard Error of the TE.

logitPc Observed baseline rate in logit scale.

N Total sample size.

Study_Design Description of the study design.

Intervention Type of vaccine used for itervention.

Valency 0 = PPV23; 1 = PPV-Other.

low_income Indicates low income patients population with 0 = no; 1 = yes.

R1 Random sequence generation (selection bias: low;high;unclear.

R2 Allocation concealment (selection bias): low;high;unclear.

R3 Confounding: low;high;unclear.

R4 Blinding of participants and personnel (performace bias): low;high;unclear.

R5 Blinding of outcome assessment (detection bias): low;high;unclear.

R6 Incomplete outcome data (attrition bias): low;high;unclear.

R7 Selective reporting (reporting bias): low;high;unclear.

Participants Comments on patients characteristics.

Source

The data were obtainded from: Moberley et al. (2013).

References

Moberley, S., Holden, J., Tatham, D., and Andrews, R. (2013), Vaccines for preventing pneumococcal infection in adults., Cochrane Database of Systematic Reviews, Issue 1. Art. No.: CD000422. DOI:10.1002/14651858.CD000422.pub3.

Verde, P.E. and Curcio, D. (2017) Hierarchical Meta-Regression Modelling: The Case of The Pneumococcal Polysaccharide Vaccine. Technical Report.

---

ppvipd *Efficacy of Pneumococcal Polysaccharide Vaccine in Preventing Invasive Pneumococcal Disease*

Description

PPV23 (23-valent pneumococcal polysaccharide vaccine) with 3 Randomized Clinical Trials; 5 Cohort Studies and 3 Case-Control Studies.

The outcome variable IPD (Invasive Pneumococcal Disease).

Format

A matrix with 11 rows and 6 columns. Each row represents study results, the columns are:

name Name of the first author and year.

TE Treatment Effect as Log Odds Ratio.

seTE Standard Error of the TE.

n.v Number of patients in the vaccination group.

n.c Number of patients in the control group.

design Description of the study design.

Source

The data were obtainded from: Falkenhorst et al. (2017).

References

Falkenhorst, G., Remschmidt, C., Harder, T., Hummers-Pradier, E., Wichmann, O., and Bogdan, C. (2017) Effectiveness of the 23-Valent Pneumococcal Polysaccharide Vaccine(PPV23) against Pneumococcal Disease in the Elderly: Systematic Review and Meta-Analysis. PLoS ONE 12(1): e0169368. doi:10.1371/journal.pone.0169368.

Verde, P.E. and Curcio, D. (2017) Hierarchical Meta-Regression Modelling: The Case of The Pneumococcal Polysaccharide Vaccine. Technical Report.

print.b3lmeta *Generic print function for b3lmeta object in jarbes.*

Description

Generic print function for b3lmeta object in jarbes.

Usage

```
## S3 method for class 'b3lmeta'
print(x, digits, ...)
```

Arguments

x	The object generated by the function b3lmeta.
digits	The number of significant digits printed. The default value is 3.
...	...

print.bcdpmeta *Generic print function for bcdpmeta object in jarbes.*

Description

Generic print function for bcdpmeta object in jarbes.

Usage

```
## S3 method for class 'bcdpmeta'
print(x, digits, ...)
```

Arguments

x	The object generated by the function bcdpmeta.
digits	The number of significant digits printed. The default value is 3.
...	...

print.bcmeta *Generic print function for bcmeta object in jarbes.*

Description

Generic print function for bcmeta object in jarbes.

Usage

```
## S3 method for class 'bcmeta'
print(x, digits, ...)
```

Arguments

x	The object generated by the function bcmeta.
digits	The number of significant digits printed. The default value is 3.
...	...

print.bcmixmeta *Generic print function for bcmixmeta object in jarbes.*

Description

Generic print function for bcmixmeta object in jarbes.

Usage

```
## S3 method for class 'bcmixmeta'
print(x, digits, ...)
```

Arguments

x	The object generated by the function bcmixmeta.
digits	The number of significant digits printed. The default value is 3.
...	...

print.bmeta *Generic print function for bcmeta object in jarbes.*

Description

Generic print function for bcmeta object in jarbes.

Usage

```
## S3 method for class 'bmeta'
print(x, digits, ...)
```

Arguments

x	The object generated by the function bcmeta.
digits	The number of significant digits printed. The default value is 3.
...	...

print.dpmeta *Generic print function for dpmeta object in jarbes.*

Description

Generic print function for dpmeta object in jarbes.

Usage

```
## S3 method for class 'dpmeta'
print(x, digits, ...)
```

Arguments

x	The object generated by the function dpmeta.
digits	The number of significant digits printed. The default value is 3.
...	...

print.dpmetareg *Generic print function for dpmeta object in jarbes.*

Description

Generic print function for dpmeta object in jarbes.

Usage

```
## S3 method for class 'dpmetareg'
print(x, digits, ...)
```

Arguments

x	The object generated by the function dpmmeta.
digits	The number of significant digits printed. The default value is 3.
...	...

print.dpmmeta *Generic print function for dpmmeta object in jarbes.*

Description

Generic print function for dpmmeta object in jarbes.

Usage

```
## S3 method for class 'dpmmeta'
print(x, digits, ...)
```

Arguments

x	The object generated by the function dpmmeta.
digits	The number of significant digits printed. The default value is 3.
...	...

print.hmr *Generic print function for hmr object in jarbes.*

Description

Generic print function for hmr object in jarbes.

Usage

```
## S3 method for class 'hmr'
print(x, digits = 3, intervals = c(0.025, 0.25, 0.5, 0.75, 0.975), ...)
```

Arguments

x	The object generated by the function hmr.
digits	The number of significant digits printed. The default value is 3.
intervals	A numeric vector of probabilities with values in [0,1]. The default value is intervals = c(0.025, 0.5, 0.975).
...	...

print.metarisk *Generic print function for metarisk object in jarbes.*

Description

Generic print function for metarisk object in jarbes.

Usage

```
## S3 method for class 'metarisk'
print(x, digits, ...)
```

Arguments

x	The object generated by the function metarisk.
digits	The number of significant digits printed. The default value is 3.
...	...

`stemcells` *Meta-analysis of 31 randomized controled trials (RCTs) with reported discrepancies*

Description

Meta-analysis of 31 randomized controled trials (RCTs) of two treatment groups of heart disease patients, where the treatment group received bone marrow stem cells and the control group a placebo treatment.

Format

A matrix with 31 rows and 11 columns. Each row represents study results, the columns are:

trial ID name of the trial.

effect.size treatment effect is measured as the difference of the ejection fraction between groups, which measures the improvement of left ventricular function in the heart.

se.effect Standard Error of the effect.size.

sample.size Total number of patients in the trial.

n.discrep Number of detected discrepancies in the published trial. Discrepancies are defined as two or more reported facts that cannot both be true because they are logically or mathematically incompatible.

Sequence Bias arising from the randomization process.

Allocation Bias due to deviations from intended interventions.

Blinding Bias introduced by lack of blinding.

Outcome Bias in measurement of the outcome.

Reporting Bias in selection of the reported result.

Other Selection bias, performance bias, detection bias, attrition bias, etc.

Source

Nowbar, A N, et al. (2014) Discrepancies in autologous bone marrow stem cell trials and enhancemen of ejection fraction (DAMASCENE): weighted regression and meta-analysis. BMJ, 348,1-9.

References

Verde, P. E. (2017) Two Examples of Bayesian Evidence Synthesis with the Hierarchical Meta-Regression Approach. Chap.9, pag 189-206. Bayesian Inference, ed. Tejedor, Javier Prieto. InTech.

summary.b3lmeta *Generic summary function for bmeta object in jarbes*

Description

Generic summary function for bmeta object in jarbes

Usage

```
## S3 method for class 'b3lmeta'
summary(object, digits = 3, ...)
```

Arguments

object	The object generated by the bmeta function.
digits	The number of significant digits printed. The default value is 3.
...	...

summary.bcdpmeta *Generic summary function for bcdpmeta object in jarbes*

Description

Generic summary function for bcdpmeta object in jarbes

Usage

```
## S3 method for class 'bcdpmeta'
summary(object, digits = 3, ...)
```

Arguments

object	The object generated by the bcmeta function.
digits	The number of significant digits printed. The default value is 3.
...	...

summary.bcmeta *Generic summary function for bcmeta object in jarbes*

Description

Generic summary function for bcmeta object in jarbes

Usage

```
## S3 method for class 'bcmeta'
summary(object, digits = 3, ...)
```

Arguments

object	The object generated by the bcmeta function.
digits	The number of significant digits printed. The default value is 3.
...	...

summary.bmeta *Generic summary function for bmeta object in jarbes*

Description

Generic summary function for bmeta object in jarbes

Usage

```
## S3 method for class 'bmeta'
summary(object, digits = 3, ...)
```

Arguments

object	The object generated by the bmeta function.
digits	The number of significant digits printed. The default value is 3.
...	...

summary.dpmeta *Generic summary function for dpmmeta object in jarbes*

Description

Generic summary function for dpmmeta object in jarbes

Usage

```
## S3 method for class 'dpmeta'
summary(object, digits = 3, ...)
```

Arguments

object	The object generated by the dmpmeta function.
digits	The number of significant digits printed. The default value is 3.
...	...

summary.dpmmeta *Generic summary function for dpmmeta object in jarbes*

Description

Generic summary function for dpmmeta object in jarbes

Usage

```
## S3 method for class 'dpmmeta'
summary(object, digits = 3, ...)
```

Arguments

object	The object generated by the dmpmeta function.
digits	The number of significant digits printed. The default value is 3.
...	...

summary.hmr *Generic summary function for hmr object in jarbes*

Description

Generic summary function for hmr object in jarbes

Usage

```
## S3 method for class 'hmr'
summary(object, digits = 3, ...)
```

Arguments

object	The object generated by the hmr function.
digits	The number of significant digits printed. The default value is 3.
...	...

summary.metarisk *Generic summary function for metarisk object in jarbes*

Description

Generic summary function for metarisk object in jarbes

Usage

```
## S3 method for class 'metarisk'
summary(object, digits = 3, ...)
```

Arguments

object	The object generated by the metarisk function.
digits	The number of significant digits printed. The default value is 3.
...	...

`trisomy21` *Meta-analysis: Observational studies assessing the relationship of a positive ICPC (Isolated Choroid Plexus Cyst) on Trisomy 21*

Description

Meta-analysis of 22 Observational Studies from PubMed, Cochrane Library and SciELO databases that assessed the relationship of a positive ICPC (Isolated Choroid Plexus Cyst) on Trisomy 21

Format

A dataframe with 22 rows and 6 columns. Each row represents study results, the columns are:

year Year of publication.

author Principal author of the publication.

y Number of cases of ICPC with Trisomy 21.

n Total number o cases with ICPC.

mean.GA Mean gestational time in weeks.

study.design Study design: prospective or retrospective cohort.

Source

Kürten C, Knippel A, Verde P, Kozlowski P. A Bayesian risk analysis for Trisomy 21 in isolated choroid plexus cyst: combining a prenatal database with a meta-analysis. J Matern Fetal Neonatal Med. 2019 Jun 11:1-9. doi: 10.1080/14767058.2019.1622666. Epub ahead of print. PMID: 31113245.

Index