Package ‘jarbes’

March 10, 2022

Type Package
Title Just a Rather Bayesian Evidence Synthesis
Version 2.0.0
Date 2022-03-08
Depends R (>= 4.0.0)
Imports rjags (>= 4.0.0), R2jags (>= 0.04-03), stats, graphics,
tidyverse, tidyr, ggplot2, ggExtra, MASS, grid, gridExtra,
mcmcplots
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 7.1.2
Encoding UTF-8
NeedsCompilation no
Author Pablo Emilio Verde [aut, cre]
Maintainer Pablo Emilio Verde <pabloemilio.verde@hhu.de>
Date/Publication 2022-03-10 11:00:12 UTC

R topics documented:

b3lmeta ................................................................. 2
bcmeta ............................................................... 4
bmeta ................................................................. 6
covid19 .............................................................. 8
diagnostic .......................................................... 9
diagnostic.b3lmeta ................................................. 10
Bayesian Meta-Analysis for Combining Studies

Description

This function performs 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,
```r
b3lmeta

nr.iterations = 10000,
nr.adapt = 1000,
nr.burnin = 1000,
nr.thin = 1,
be.quiet = FALSE,
r2jags = TRUE
)

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 adaption.

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.

r2jags Which interface is used to link R to JAGS (rjags and R2jags), default value is R2Jags=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


Examples

## Not run:
library(jarbes)

## End(Not run)

### Description

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

### 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,
data
mean.mu
sd.mu
scale.sigma.between
df.scale.between
b.0
b.1
nu
nu.estimate
nr.chains
nr.iterations
nr.adapt
nr.burnin

Arguments
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.
Prior mean of the overall mean parameter mu, default value is 0.
Prior standard deviation of mu, the default value is 10.
Prior scale parameter for scale gamma distribution for the precision between
studies. The default value is 0.5.
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.
Lower bound of the bias parameter B, the default value is 0.
Upper bound of the bias parameter B, the default value is 10.
Parameter for the prior Beta distribution for the probability of bias. Default
value is a0 = 1.
Parameter for the prior Beta distribution for the probability of bias. Default
value is a1 = 1.
Parameter for the Beta distribution for the quality weights. The default value is
nu = 0.5.
If TRUE, then we estimate nu from the data.
If nu.estimate = TRUE, this parameter is the shape parameter of the prior Gamma
distribution for nu.
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.
Number of chains for the MCMC computations, default 2.
Number of iterations after adapting the MCMC, default is 10000. Some models
may need more iterations.
Number of iterations in the adaptation process, defualt is 1000. Some models
may need more iterations during adptation.
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.
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.
r2jags   Which interface is used to link R to JAGS (rjags and R2jags), default value is R2Jags=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 "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


Examples

```r
## Not run:
library(jarbes)
# Example ppvipd data
data(ppvipd)

## End(Not run)
```

---

**bmety**

*Bayesian Meta-Analysis for Combining Studies*

Description

This function performs 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,
  r2jags = TRUE
)
```

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 adaptation.
- **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.
- **r2jags**: Which interface is used to link R to JAGS (rjags and R2jags), default value is R2jags=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


Examples

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

# Example: 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)
```

Meta-analysis: Observational studies assessing the impact of risk factors on the severity and mortality of COVID-19 cases
### Description


### 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**: Endpoint: 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


---

```r
# diagnostic

### 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 performs an approximated Bayesian cross-validation for a b3lmeta object.

**Usage**

```r
## 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 bcmeta**  

*Diagnostic function for bcmeta object in jarbes*

**Description**

This function performs an approximated Bayesian cross-validation for a bcmeta object and specially designed diagnostics to detect the existence of a biased component.
### S3 method for class 'bcmeta'

```r
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.
**diagnostic.bmeta**

Diagnostic function for bmeta object in jarbes

**Description**

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

**Usage**

```r
## 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.
Diagnostic function for hmr object in jarbes

Description

This function performs a specially designed diagnostic for a hmr object.

Usage

```r
## S3 method for class 'hmr'
diagnostic(
  object,
  median.w = 1.5,
  study.names = character(0),
  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,
  ...
)
```

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.
- **mu.phi**: Also plot the distribution 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.
- **...**: Further arguments.
### diagnostic.metarisk  
*Diagnostic function for metarisk object in jarbes*

#### Description

This function performs a specially designed diagnostic for a metarisk object.

#### Usage

```r
## 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.
- **...**: Additional arguments.

---

### 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


References


---

**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 lesions 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**

**References**

---

**hmr**  
*Bayesian meta-analysis for cross-design synthesis.*

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

**Usage**
```r
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,
be.quiet = FALSE,
r2jags = TRUE
)

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 characteristics.
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 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.
sd.mu.phi
Prior standard deviation of mu.phi, default value is 1.
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. This 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 metadiag can be extracted with R2jags or with rjags. In addition a summary, a print and a plot function are implemented for this type of object.

References


Examples

## Not run:
# Examples of new plot and diagnostic functions for version >= 2.0.0

# Example: from Verde 2019, Section 5

library(jarbes)

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

data("healingipd")


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)

summary(mx2)
plot(mx2, names = c("PAD", "dialysis", "male"))
diagnostic(mx2)
diagnostic(mx2, mu.phi = FALSE, study.names = healing$Study)
diagnostic(mx2, study.names = healing$Study)
betaplot(mx2)

# This experiment corresponds to Section 4 in Verde (2019).
# Experiment: Combining aggregated 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 difficult 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)
library(R2jags)

# Simulation of the IPD data
invlogit <- function (x)
{
  1/(1 + exp(-x))
}

# Data set for mu.phi = 0.5 ..........................
# Parameters Aggregated Data:

# mean control
pc <- 0.7
# mean treatment
pt <- 0.4
# reduction of "amputations" odds ratio
OR <- (pt/(1-pt)) /(pc/(1-pc))
# mu_2: treatment effect ...
log(OR)
mu.2.true <- log(OR)
# mu_1
mu.1.true <- log(pc/(1-pc)) # Baseline risk
mu.1.true
#sigma_1 # Between studies variability
sigma.1.true <- 1
#sigma_2
sigma.2.true <- 0.5
# rho: correlation between treatment effect and baseline risk
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)

# Parameters values IPD

# 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)

# Aggregated Data

# Experiment 1: External validity bias
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:
nc <- round(runif(35, min = 30, max = 60),0)
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
\[ n_{21} \leftarrow AD.s1nt - AD.s1yt \]
\[ n_{22} \leftarrow AD.s1nc - AD.s1yc \]
\[ AD.s1TE \leftarrow \log\left(\frac{(n_{11} + incr.e) \cdot (n_{22} + incr.c)}{(n_{12} + incr.e) \cdot (n_{21} + incr.c)}\right) \]
\[ AD.s1seTE \leftarrow \sqrt{\frac{1}{(n_{11} + incr.e)} + \frac{1}{(n_{12} + incr.e)} + \frac{1}{(n_{21} + incr.c)} + \frac{1}{(n_{22} + incr.c)}} \]
\[ Pc \leftarrow \frac{(n_{12} + incr.c)}{(AD.s1nc + 2\cdot incr.c)} \]
\[ AD.s1$\logitPc \leftarrow \log\left(\frac{Pc}{1-Pc}\right) \]
\[ AD.s1$Ntotal \leftarrow AD.s1nc + AD.s1nt \]
\[ rm(list=c("Pc", "n_{11}", "n_{12}", "n_{21}", "n_{22}", "incr.c", "incr.e")) \]

# 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)
```r
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 = 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()
```

## metarisk

Bayesian meta-analysis for external validity using baseline risk

```r
library(jags)

# Catter plots for regression coefficients ......................
library(rjags)

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

# End of the examples.
```

## metarisk

Bayesian meta-analysis for external validity using baseline risk
Description

This function performs 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

```r
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,
  be.quiet = FALSE,
  r2jags = TRUE
)
```

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.
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.

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.

Upper bound of the uniform prior of sigma.1, default value is 5.

Upper bound of the uniform prior of sigma.2, default value is 5.

Mean of rho in the Fisher scale default value is 0.

Standard deviation of rho in the Fisher scale, default value is 1/sqrt(2).

If de.estimate = FALSE, then df is the degrees of freedom for the scale mixture distribution, default value is 4.

Estimate the posterior of df. The default value is FALSE.

Lower bound of the prior of df. The default value is 3.

Upper bound of the prior of df. The default value is 30.

Split the w parameter in two independent weights one for each random effect. The default value is FALSE.

Number of chains for the MCMC computations, default 5.

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

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

Number of iteration discarded for burnin period, default is 1000. Some models may need a longer burnin period.

Thinning rate, it must be a positive integer, the default value is 1.

Do not print warning message if the model does not adapt default value is FALSE. If you are not sure about the adaptation period choose be.quiet=TRUE.

Which interface is used to link R to JAGS (rjags and R2jags) default value is R2Jags=TRUE.

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**


**Examples**

```r
## 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 iterations.
# For example use: nr.burnin = 5000 and nr.iterations = 20000

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 <- \[(\text{pt}/(1-\text{pt})) / (\text{pc}/(1-\text{pc}))\]

# 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))

# 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)

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

eta.true <- mu.2.true + sigma.2.true/sigma.1.true * rho.true* mu.T

exp(eta.true)

# 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$yt
n12 <- AD.s1$syc
n21 <- AD.s1$nt - AD.s1$yt
n22 <- AD.s1$nc - AD.s1$syc
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 assessing 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 bcmeta object in jarbes.**

**Description**

Generic plot function for bcmeta object in jarbes.

Generic plot function for b3lmeta object in jarbes.

**Usage**

```r
## 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",
  ...
)
```

```r
## 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.bmeta**

*Generic plot function for bcmeta object in jarbes.*

---

**Description**

Generic plot function for bcmeta object in jarbes.

**Usage**

```r
## 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

```r
## 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.hmr**

Generic plot function for hmr object in jarbes.

Description

Generic plot function for hmr object in jarbes.

Usage

```r
## S3 method for class 'hmr'
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",
  names = NULL,
  name.side = NULL,
  ...,
)
```
plot.metarisk

Generic plot function for metarisk object in jarbes.

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.
- `names`: Add IPD names to the plot.
- `name.side`: Set the side of each name in the plot relative to the vertical line.
- `...`: ...

Usage

```r
## 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.
- `...`: ...
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 intervention.
- **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 (performance 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.
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 obtained from: Falkenhorst et al. (2017).

References


Generic print function for bcmeta object in jarbes.

**Usage**

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

**Arguments**

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

Generic print function for bcmeta object in jarbes.

**Usage**

```r
## 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.bmeta**  
*Generic print function for bcmeta object in jarbes.*

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

**Usage**  
```r  
## 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.hmr**  
*Generic print function for hmr object in jarbes.*

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

**Usage**  
```r  
## 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

```r
## 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 controlled trials (RCTs) with reported discrepancies

Description

Meta-analysis of 31 randomized controlled 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 5 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.

Source

References

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.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 of cases with ICPC.
- mean.GA: Mean gestational time in weeks.
- study.design: Study design: prospective or retrospective cohort.

Source
Index

* datasets
  - covid19, 8
  - healing, 14
  - healingipd, 15
  - ppvcap, 37
  - ppvipd, 38
  - stemcells, 41
  - trisomy21, 44

b3lmeta, 2
bcmeta, 4
bmeta, 6
covid19, 8
diagnostic, 9
diagnostic.b3lmeta, 10
diagnostic.bcmeta, 10
diagnostic.bmeta, 12
diagnostic.hmr, 13
diagnostic.metarisk, 14

healing, 14
healingipd, 15
hmr, 16

metarisk, 25

plot.b3lmeta, 33
plot.bcmeta, 34
plot.bmeta, 34
plot.hmr, 35
plot.metarisk, 36
ppvcap, 37
ppvipd, 38
print.b3lmeta, 39
print.bcmeta, 39
print.bmeta, 40
print.hmr, 40
print.metarisk, 41

stemcells, 41
summary.b3lmeta, 42
summary.bcmeta, 42
summary.bmeta, 43
summary.hmr, 43
summary.metarisk, 44

trisomy21, 44