

Package ‘jarbes’

March 11, 2019

Type Package

Title Just a Rather Bayesian Evidence Synthesis

Version 1.7.2

Date 2019-03-11

Depends R (>= 3.4.0)

Imports rjags (>= 3.4.0), R2jags (>= 0.04-03), stats, graphics, ggplot2, ggExtra, MASS, grid, gridExtra, mcmcplots

SystemRequirements JAGS (>= 3.4.0) (see <http://mcmc-jags.sourceforge.net>)

Description Provides a new class of Bayesian meta-analysis models that we called “The Hierarchical Meta-Regression” (HMR). The aim of HMR is to incorporate into the meta-analysis, the data collection process, which results in a model for the internal and external validity bias. In this way, it is possible to combine studies of different types. For example, we can combine the results of randomized control trials (RCTs) with the results of observational studies (OS). The statistical methods and their applications are described in Verde (2019) <doi:10.1002/bimj.201700266> and in Verde (2017) <doi:10.5772/intechopen.70231>.

License GPL (>= 2)

Repository CRAN

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation no

Author Pablo Emilio Verde [aut, cre]

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

Date/Publication 2019-03-11 16:50:03 UTC

R topics documented:

ges	2
gesmix	4
healing	6
healingipd	7

hmr	8
metarisk	17
plot.ges	24
plot.gesmix	24
plot.hmr	25
plot.metarisk	25
ppvcap	26
ppvipd	27
print.ges	28
print.gesmix	28
print.hmr	29
print.metarisk	29
stemcells	30
summary.ges	30
summary.gesmix	31
summary.hmr	31
summary.metarisk	32

Index 33

ges	<i>Generalized Evidence Synthesis with Direct Penalization of Observational Studies</i>
-----	---

Description

This function fits a hierarchical meta-regression model based on a three levels random-effects model with direct penalization of observational studies.

Usage

```
ges(data, EmBi = FALSE, mu.bias = -1.25, sigma.bias = 0.1,
     ExPe = FALSE, K = 0.5, df = 4, mean.mu.0 = 0, sd.mu.0 = 10,
     scale.sigma.within = 0.5, scale.sigma.between = 0.5,
     df.scale.within = 1, df.scale.between = 1, nr.chains = 2,
     nr.iterations = 20000, nr.adapt = 1000, nr.burnin = 1000,
     nr.thin = 1, be.quiet = FALSE, r2jags = TRUE)
```

Arguments

data	A data frame where the first column is the treatment effect TE, the second column is the standard error of the treatment effect seTE and the third column is a factor indicating the statistical design of the study.
EmBi	Indicates if Empirical Bias Penalization is performed. The default is FALSE.
mu.bias	Empirical mean bias. The default value is -1.25 (BRANDO study, Savovic et al. 2012).
sigma.bias	Empirical bias sd. The default value is 0.1 (BRANDO study, Savovic et al. 2012).

ExPe	Indicates if Explicit Bias Penalization is performed. The default is FALSE.
K	Prior expected value of the weights used for observational studies, $E(w_{OS})=K$. The default value is 0.5.
df	The degrees of freedom for the scale mixture distribution. The default value is 4.
mean.mu.0	Prior mean of the overall mean parameter mu.0. The default value is 0.
sd.mu.0	Prior standard deviation of mu.0
scale.sigma.within	Prior scale parameter for the standard deviation within study type. The default value is 0.5.
scale.sigma.between	Prior scale parameter for the standard deviation between study type. The default value is 0.5.
df.scale.within	Degrees of freedom of the standard deviation within study type. The default value is 1.
df.scale.between	Degrees of freedom of the standard deviation between study type. The default value is 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, default 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.
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.

Value

This function returns an object of the class "ges". This object contains the MCMC output of each parameter and hyper-parameter in the model, the data frame used for fitting the model and type of random effects distribution

The results of the object of the class ges 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. (2017) Hierarchical Meta-Regression Modelling: The Case of The Pneumococcal Polysaccharide Vaccine. Technical Report.

Examples

```
## Not run:

library(jarbes)
data(ppvipv)

# Three levels hierarchical model ....
m1.ges = ges(ppvipv)
summary(m1.ges)

# Three levels hierarchical mode with empirical bias penalization...
m2.ges = ges(ppvipv, EmBi = TRUE)
summary(m2.ges)

# Three levels hierarchical model with explicit bias penalization ...
m3.ges = ges(ppvipv, ExPe = TRUE, K = 0.5, df = 10)
summary(m3.ges)

## End(Not run)
```

gesmix

Generalized Evidence Synthesis with Indirect Penalization of Observational Studies

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

```
gesmix(data, mean.mu = 0, sd.mu = 10^6, scale.sigma = 0.5,
       K.lower = -10, K.upper = 0, 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 three variables with the following names: 1) TE = treatment effect, 2) seTE = the standard error of the treatment effect 3) design = the study type. The other columns of the data frame may correspond to the study name and other covariates that characterize the study and could influence internal validity bias (e.g. evaluation of risk of bias).
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^{-6} .

<code>scale.sigma</code>	Prior scale parameter for the standard deviation between studies, the default value is 0.5.
<code>K.lower</code>	Lower bound of the bias parameter K, the default value is -10.
<code>K.upper</code>	Upper bound of the bias parameter K, the default value is 0.
<code>nr.chains</code>	Number of chains for the MCMC computations, default 2.
<code>nr.iterations</code>	Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.
<code>nr.adapt</code>	Number of iterations in the adaptation process, default is 1000. Some models may need more iterations during adptation.
<code>nr.burnin</code>	Number of iteration discared for burnin period, default is 1000. Some models may need a longer burnin period.
<code>nr.thin</code>	Thinning rate, it must be a positive integer, the default value 1.
<code>be.quiet</code>	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 <code>be.quiet=TRUE</code> .
<code>r2jags</code>	Which interface is used to link R to JAGS (<code>rjags</code> and <code>R2jags</code>), default value is <code>R2Jags=TRUE</code> .

Details

The results of the object of the class `gesmix` 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 "`gesmix`". 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. and Curcio, D. (2019) Hierarchical Meta-Regression Modelling: The Case of The Pneumococcal Polysaccharide Vaccine. Technical Report.

Examples

```
## Not run:

library(jarbes)

out = gesmix(data = ppvipd)
attach.jags(out, overwrite = TRUE)

par(mfrow = c(1,2))
hist(lambda[,1], breaks = 100, probability = TRUE,
      ylim = c(0, 3),
      xlim = c(-2, 0.85), main = "",
```

```

xlab = "Treatment Effect: log(Odds Ratio)")

lines(density(lambda[,2]), lwd = 3, col = "red")
lines(density(lambda[,1]), lwd = 3, col = "blue")
abline(v = 0, lty = 2)
legend(-2, 3, legend = c("RCTs' Component",
                        "Obsevationals' Component"),
      col = c("blue", "red"),
      lty = rep(1, 2),
      lwd = rep(2, 2))
hist(c(lambda[,1],lambda[,2]+0.337),
     breaks = 150,
     probability = TRUE,
     ylim = c(0, 3.5),
     xlim = c(-2.1, 0.2), main = "",
     xlab = "Treatment Effect: log(Odds Ratio)")
lines(density(c(lambda[,1],lambda[,2]+0.337)), col = "blue", lwd =3)
abline(v = quantile(c(lambda[,1],lambda[,2]+0.337),
                    probs = c(0.025,0.5,0.975)), col = "blue",
      lty = 2)
m.pool <- log(0.43)
sd.pool <- (log(0.54) - log(0.34))/(2*1.96)
curve(dnorm(x, m = m.pool, s = sd.pool),
      from = -3, to = 0.5, add = TRUE, col = "black", lwd =3)
legend(-2.1, 3,
      legend = c("Bias-adjusted",
                "Biased result"),
      col = c("blue", "black"),
      lty = rep(1, 2),
      lwd = rep(2, 2))
abline(v = 0, lty = 2)
arrows(-1.5, 1.1, -1, 1.55, lwd =2, length=0.1, angle=20)
text(-1.8, 1, "Ignoring study types")
par(mfrow = c(1,1))

## End(Not run)

```

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 obtained 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	<i>Individual participant data for diabetic patients</i>
------------	--

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.

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

```
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

<code>data</code>	Aggregated data results: a data frame where the first four columns containing the number of events in the control group (<code>yc</code>), the number of patients in the control group (<code>nc</code>), the number of events in the treatment group (<code>yt</code>) and the number of patients in the treatment group (<code>nt</code>). If <code>two.by.two = TRUE</code> a data frame where each line contains the trial results with column names: <code>yc</code> , <code>nc</code> , <code>yt</code> , <code>nt</code> .
<code>two.by.two</code>	If TRUE indicates that the trial results are with names: <code>yc</code> , <code>nc</code> , <code>yt</code> , <code>nt</code> .
<code>dataIPD</code>	Individual participant data: a data frame where the first column is the outcome variable and the other columns represent individual participant characteristics.
<code>re</code>	Random effects distribution for the resulting model. Possible values are <i>normal</i> for bivariate random effects and <i>sm</i> for scale mixtures.
<code>link</code>	The link function used in the model. Possible values are <i>logit</i> , <i>cloglog</i> <i>probit</i> .
<code>mean.mu.1</code>	Prior mean of baseline risk, default value is 0.
<code>mean.mu.2</code>	Prior mean of treatment effect, default value is 0.
<code>mean.mu.phi</code>	Prior mean of the bias parameter which measures the difference between the baseline mean <code>mu.1</code> and the intercept parameter of the logistic regression of the individual participant data. The default value is 0.
<code>sd.mu.1</code>	Prior standard deviation of <code>mu.1</code> , default value is 1. The default prior of <code>mu.1</code> is a logistic distribution with mean 0 and dispersion 1. The implicit prior for <code>mu.1</code> in the probability scale is a uniform between 0 and 1.
<code>sd.mu.2</code>	Prior standard deviation of <code>mu.2</code> , default value is 1. The default prior of <code>mu.2</code> is a logistic distribution with mean 0 and dispersion 1. The implicit prior for <code>mu.2</code> in the probability scale is a uniform between 0 and 1.
<code>sd.mu.phi</code>	Prior standard deviation of <code>mu.phi</code> , default value is 1.
<code>sigma.1.upper</code>	Upper bound of the uniform prior of <code>sigma.1</code> , default value is 5.
<code>sigma.2.upper</code>	Upper bound of the uniform prior of <code>sigma.2</code> , default value is 5.
<code>sigma.beta.upper</code>	Upper bound of the uniform prior of <code>sigma.beta</code> , default value is 5.
<code>mean.Fisher.rho</code>	Mean of <code>rho</code> in the Fisher scale, default value is 0.
<code>sd.Fisher.rho</code>	Standard deviation of <code>rho</code> in the Fisher scale, default value is $1/\sqrt{2}$.
<code>df</code>	If <code>de.estimate = FALSE</code> , then <code>df</code> is the degrees of freedom for the scale mixture distribution, default value is 4.
<code>df.estimate</code>	Estimate the posterior of <code>df</code> . The default value is FALSE.
<code>df.lower</code>	Lower bound of the prior of <code>df</code> . The default value is 3.
<code>df.upper</code>	Upper bound of the prior of <code>df</code> . The default value is 30.
<code>split.w</code>	Split the <code>w</code> parameter in two independent weights one for each random effect. The default value is FALSE.
<code>nr.chains</code>	Number of chains for the MCMC computations, default 5.
<code>nr.iterations</code>	Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.

<code>nr.adapt</code>	Number of iterations in the adaptation process, default is 1000. Some models may need more iterations during adptation.
<code>nr.burnin</code>	Number of iteration discarded for burnin period, default is 1000. Some models may need a longer burnin period.
<code>nr.thin</code>	Thinning rate, it must be a positive integer, the default value 1.
<code>be.quiet</code>	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 <code>be.quiet=TRUE</code> .
<code>r2jags</code>	Which interface is used to link R to JAGS (<code>rjags</code> and <code>R2jags</code>) default value is <code>R2Jags TRUE</code> .

Details

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 $\geq 3.3.0$, 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

- 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. (2019) The hierarchical meta-regression approach and learning from clinical evidence. *Biometrical Journal*. 1 - 23.

Examples

```
## Not run:

# Example: from Verde 2019, Section 5
```

```
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 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 identified. 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:
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"))

# 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)

# 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)

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 = TRUE,
          labels.loc="axis", cex =0.7,
          style = "plain",reorder = FALSE,
          denstrip = FALSE)

caterplot(mcmc.x.3,
          parms = par.names,
          col = "grey", lty = 2,
          labels = greek.names,
          greek = TRUE,
          labels.loc="axis", cex =0.7,
          style = "plain", reorder = FALSE,
          denstrip = FALSE,
          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 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

```

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 <i>normal</i> for bivariate random effects and <i>sm</i> for scale mixtures.
link	The link function used in the model. Possible values are <i>logit</i> , <i>cloglog</i> <i>probit</i> .
mean.mu.1	Prior mean of baseline risk, default value is 0.
mean.mu.2	Prior mean of 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.

<code>sd.mu.2</code>	Prior standard deviation of <code>mu.2</code> , default value is 1. The default prior of <code>mu.2</code> is a logistic distribution with mean 0 and dispersion 1. The implicit prior for <code>mu.2</code> in the probability scale is a uniform between 0 and 1.
<code>sigma.1.upper</code>	Upper bound of the uniform prior of <code>sigma.1</code> , default value is 5.
<code>sigma.2.upper</code>	Upper bound of the uniform prior of <code>sigma.2</code> , default value is 5.
<code>mean.Fisher.rho</code>	Mean of <code>rho</code> in the Fisher scale default value is 0.
<code>sd.Fisher.rho</code>	Standard deviation of <code>rho</code> in the Fisher scale, default value is $1/\sqrt{2}$.
<code>df</code>	If <code>de.estimate = FALSE</code> , then <code>df</code> is the degrees of freedom for the scale mixture distribution, default value is 4.
<code>df.estimate</code>	Estimate the posterior of <code>df</code> . The default value is <code>FALSE</code> .
<code>df.lower</code>	Lower bound of the prior of <code>df</code> . The default value is 3.
<code>df.upper</code>	Upper bound of the prior of <code>df</code> . The default value is 30.
<code>split.w</code>	Split the <code>w</code> parameter in two independent weights one for each random effect. The default value is <code>FALSE</code> .
<code>nr.chains</code>	Number of chains for the MCMC computations, default 5.
<code>nr.iterations</code>	Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.
<code>nr.adapt</code>	Number of iterations in the adaptation process, default is 1000. Some models may need more iterations during adaptation.
<code>nr.burnin</code>	Number of iteration discarded for burnin period, default is 1000. Some models may need a longer burnin period.
<code>nr.thin</code>	Thinning rate, it must be a positive integer, the default value is 1.
<code>be.quiet</code>	Do not print warning message if the model does not adapt default value is <code>FALSE</code> . If you are not sure about the adaptation period choose <code>be.quiet=TRUE</code> .
<code>r2jags</code>	Which interface is used to link R to JAGS (<code>rjags</code> and <code>R2jags</code>) default value is <code>R2Jags=TRUE</code> .

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

```
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

data(ppvcap)
dat <- ppvcap[, c("yt","nt","yc","nc")]
res.hmr.1 <- metarisk(dat, # Data frame
                     two.by.two = FALSE, # Data is given as: (yt, nt, yc, nc)
                     re = "sm", # Random effects distribution
                     link = "logit", # Link function
                     sd.Fisher.rho = 1.7, # Prior standard deviation of correlation
                     split.w = TRUE, # Split the studies' weights
                     df.estimate = TRUE, # Estimate the posterior of df parameter.
                     nr.burnin = 100, # Iterations for burnin
                     nr.iterations = 1000, # Total iterations
                     nr.chains = 2, # Number of chains
                     r2jags = TRUE) # Use r2jags as interface to jags

print(res.hmr.1, digits = 3)
```

```
# The following examples corresponds to Section 4 in Verde (2018).
# 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 partical value".
#   Note that df = 1 fits a Cauchy bivariate distribution to
#   the random effects.
#End of the examples
```

plot.ges	<i>Generic plot function for ges object in jarbes.</i>
----------	--

Description

Generic plot function for ges object in jarbes.

Usage

```
## S3 method for class 'ges'
plot(x, ...)
```

Arguments

x	The object generated by the ges function.
...	...

plot.gesmix	<i>Generic plot function for metamixt object in jarbes.</i>
-------------	---

Description

Generic plot function for metamixt object in jarbes.

Usage

```
## S3 method for class 'gesmix'
plot(x, ...)
```

Arguments

x	The object generated by the metamixt function.
...	...

plot.hmr	<i>Generic plot function for hmr object in jarbes.</i>
----------	--

Description

Generic plot function for hmr object in jarbes.

Usage

```
## S3 method for class 'hmr'  
plot(x, ...)
```

Arguments

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

plot.metarisk	<i>Generic plot function for metarisk object in jarbes.</i>
---------------	---

Description

Generic plot function for metarisk object in jarbes.

Usage

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

Arguments

x	The object generated by the metarisk function.
...	...

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 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.

Source

The data were obtained 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 obtained 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.ges	<i>Generic print function for ges object in jarbes.</i>
-----------	---

Description

Generic print function for ges object in jarbes.

Usage

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

Arguments

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

print.gesmix	<i>Generic print function for metamixt object in jarbes.</i>
--------------	--

Description

Generic print function for metamixt object in jarbes.

Usage

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

Arguments

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

print.hmr	<i>Generic print function for hmr object in jarbes.</i>
-----------	---

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	<i>Generic print function for metarisk object in jarbes.</i>
----------------	--

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	<i>Meta-analysis of 31 randomized controlled trials (RCTs) with reported discrepancies</i>
-----------	--

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

Nowbar, A N, et al. (2014) Discrepancies in autologous bone marrow stem cell trials and enhancement 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.ges	<i>Generic summary function for ges object in jarbes</i>
-------------	--

Description

Generic summary function for ges object in jarbes

Usage

```
## S3 method for class 'ges'
summary(object, digits = 3, intervals = c(0.025, 0.5,
0.975), ...)
```

Arguments

object	The object generated by the ges function.
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).
...	...

summary.gesmix	<i>Generic summary function for metamixt object in jarbes</i>
----------------	---

Description

Generic summary function for metamixt object in jarbes

Usage

```
## S3 method for class 'gesmix'
summary(object, digits = 3, intervals = c(0.025, 0.5,
0.975), ...)
```

Arguments

object	The object generated by the metamixt function.
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).
...	...

summary.hmr	<i>Generic summary function for hmr object in jarbes</i>
-------------	--

Description

Generic summary function for hmr object in jarbes

Usage

```
## S3 method for class 'hmr'
summary(object, digits = 3, intervals = c(0.025, 0.5,
0.975), ...)
```

Arguments

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

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, intervals = c(0.025,
  0.5, 0.975), ...)
```

Arguments

object	The object generated by the metarisk function.
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).
...	...

Index

*Topic **datasets**

- healing, [6](#)
- healingipd, [7](#)
- ppvcap, [26](#)
- ppvipd, [27](#)
- stemcells, [30](#)

- ges, [2](#)
- gesmix, [4](#)

- healing, [6](#)
- healingipd, [7](#)
- hmr, [8](#)

- metarisk, [17](#)

- plot.ges, [24](#)
- plot.gesmix, [24](#)
- plot.hmr, [25](#)
- plot.metarisk, [25](#)
- ppvcap, [26](#)
- ppvipd, [27](#)
- print.ges, [28](#)
- print.gesmix, [28](#)
- print.hmr, [29](#)
- print.metarisk, [29](#)

- stemcells, [30](#)
- summary.ges, [30](#)
- summary.gesmix, [31](#)
- summary.hmr, [31](#)
- summary.metarisk, [32](#)