Package 'bamdit'

October 12, 2022

Type Package

Title Bayesian Meta-Analysis of Diagnostic Test Data

Version 3.4.0

Date 2022-04-04

Depends R (>= 4.0.0)

Imports rjags (>= 4.0.0), R2jags (>= 0.04-03), stats, ggplot2, ggExtra, MASS, grid, gridExtra

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 2022-04-04 13:40:02 UTC

R topics documented:

`bamdit-package` *Bayesian Meta-Analysis of Diagnostic Test Data*

Description

Bayesian meta-analysis of diagnostic test data based on a scale mixtures bivariate random-effects model. This package was developed with the aim of simplifying the use of meta-analysis models that up to now have demanded great statistical expertise in Bayesian meta-analysis. The package implements a series of innovative statistical techniques including: the BSROC (Bayesian Summary ROC) curve, the BAUC (Bayesian AUC), predictive surfaces, the use of prior distributions that avoid boundary estimation problems of component of variance and correlation parameters, analysis of conflict of evidence and robust estimation of model parameters. In addition, the package comes with several published examples of meta-analysis that can be used for illustration or further research in this area.

Details

Package:	bamdit
Type:	Package
Version:	3.4.0
Date:	2022-04-04
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

PD Dr. Pablo Emilio Verde <pabloemilio.verde@hhu.de>

References

Verde P. E. (2010). Meta-analysis of diagnostic test data: A bivariate Bayesian modeling approach. Statistics in Medicine. 29(30):3088-102. doi: 10.1002/sim.4055.

Verde P. E. (2018). bamdit: An R Package for Bayesian Meta-Analysis of Diagnostic Test Data. Journal of Statistical Software. Volume 86, issue 10, pages 1–32.

`bsroc` *bsroc*

Description

This function plots the observed data in the ROC (Receiving Operating Charachteristics) space with the Bayesian SROC (Summary ROC) curve. The predictive curves are approximated using a parametric model.

Usage

```
bsroc(
  m,
  level = c(0.05, 0.5, 0.95),
  title = "Bayesian SROC Curve",
  fpr.x = seq(0.01, 0.95, 0.01),
  partial.AUC = TRUE,
  xlim.bsroc = c(0, 1),
  ylim.bsroc = c(0, 1),
  lower.auc = 0,
  upper.auc = 0.95,
  col.fill.points = "blue",
  results.bauc = TRUE,
  results.bsroc = FALSE,
  plot.post.bauc = FALSE,
  bins = 30,
  scale.size.area = 10
)
```

Arguments

`m`	The object generated by metadiag.
`level`	Credibility levels of the predictive curve
`title`	Optional parameter for setting a title in the plot.
`fpr.x`	Grid of values where the conditionl distribution is calculated.
`partial.AUC`	Automatically calculate the AUC for the observed range of FPRs, default is TRUE.
`xlim.bsroc`	Graphical limits of the x-axis for the BSROC curve plot.
`ylim.bsroc`	Graphical limits of the y-axis for the BSROC curve plot.

lower.auc	Lower limit of the AUC.
upper.auc	Upper limit of the AUC.
col.fill.points	Color used to fill points, default is blue.
results.bauc	Print results of the Bayesian Area Under the Curve, default value is TRUE.
results.bsroc	Print results of the Bayesian SROC curve, default value is FALSE.
plot.post.bauc	The BSROC and the posterior of the BAUC are ploted in the same page, default is FALSE.
bins	Histograms' bins.
scale.size.area	Scale area for the ploted points, default = 10.

References

Verde P. E. (2010). Meta-analysis of diagnostic test data: A bivariate Bayesian modeling approach. Statistics in Medicine. 29(30):3088-102. doi: 10.1002/sim.4055.

Verde P. E. (2018). bamdit: An R Package for Bayesian Meta-Analysis of Diagnostic Test Data. Journal of Statisticsl Software. Volume 86, issue 10, pages 1–32.

See Also

metadiag.

Examples

```
## execute analysis
## Not run:
# Example: data from Glas et al. (2003).....................................

data(glas)
glas.t <- glas[glas$marker == "Telomerase", 1:4]
glas.m1 <- metadiag(glas.t, re = "normal", link = "logit")
bsroc(glas.m1)
bsroc(glas.m1, plot.post.bauc = TRUE)

# Example: data from Scheidler et al. (1997)
# In this example the range of the observed FPR is less than 20%.
# Calculating the BSROC curve makes no sense! You will get a warning message!

data(mri)
mri.m <- metadiag(mri)
bsroc(mri.m)

## End(Not run)
```

ct *Diagnosis of appendicities with computer tomography scans*

Description

This data frame corresponds to 51 clinical studies reporting the accuracy of computer tomography (CT) scans for the diagnosis of appendicities.

Format

A matrix with 51 rows and 17 columns. Each row represents study results, the columns are:

tp number of true positives.

n1 number of patients with disease.

fp number of false positives.

n2 number of patients without disease.

Author First author and year.

country Country: EU = 1, others/USA = 2.

hosp Type of hospital: 1 = university, 2 = others.

inclus Inclusion criteria: 1 = Suspected, 2 = appendectomy.

indfind Other CT findings included: 1 = no, 2 = yes.

design Study design: 1 = prospective, 2 = retrospective.

contr Contrast medium: 1 = no, 2 = yes.

localis Localisation: 1 = one area, 2 = more than one area.

child Children included: 1 = no, 2 = yes.

fup.na Followup: 0 = no, 1 = yes.

refer.na Valid reference: 0 = no, 1 = yes.

sample.na Sample: 0 = selected, 1= consecutive/random.

gender.na Gender, female: 0 = less than 50%; 1 = more than 50%.

Details

This data frame corresponds to 51 clinical studies reporting the accuracy of computer tomography (CT) scans for the diagnosis of appendicities.

Source

The data were obtainded from

Ohmann C, Verde PE, Gilbers T, Franke C, Fuerst G, Sauerland S, Boehner H. (2006) Systematic review of CT investigation in suspected acute appendicitis. *Final Report; Coordination Centre for Clinical Trials, Heinrich-Heine University*. Moorenstr. 5, D-40225 Duesseldorf Germany.

References

Verde P. E. (2010). Meta-analysis of diagnostic test data: A bivariate Bayesian modeling approach. *Statistics in Medicine*. **29**, 3088-3102.

`diabetes` *Systematic review which compares the accuracy of HbA1c vs FPG in diabetes*

Description

This data frame contains results of diagnostic accuracy of 38 studies which reported comparison of sensitivity and specificity between HbA1c vs FPG in a population based screening for type 2 diabetes.

Format

A data frame with 38 rows and 9 columns. Each row represents study results, the columns are:

Study Name of the first author.

TP_HbA1c Number of true positive cases for HbA1c.

FP_HbA1c Number of false positive cases for HbA1c.

FN_HbA1c Number of false negative cases for HbA1c.

TN_HbA1c Number of true negative cases for HbA1c.

TP_FPG Number of true positive cases for FPG.

FP_FPG Number of false positive cases for FPG.

FN_FPG Number of false negative cases for FPG.

TN_FPG Number of true negative cases for FPG.

Details

This data frame contains results of diagnostic accuraccy of 38 studies which reported comparison of sensitivity and specificity between HbA1c vs FPG in a population-based screening for type 2 diabetes.

Source

Hoyer, A., Kuss, O. Meta-analysis for the comparison of two disgnostic test: a new approach based on copulas. Stat. Med. 2018; 37:739-748

ep *Ectopic pregnancy vs. all other pregnancies data*

Description

Ectopic pregnancy vs. all other pregnancies data Table III Mol et al. 1998

Format

A matrix with 21 rows and 8 columns. Each row represents study results, the columns are:

tp number of true positives.

n1 number of patients with disease.

fp number of false positives.

n2 number of patients without disease.

d1 Prospective vs. retrospective.

d2 Cohort vs. case-control

d3 Consecutive sampling patients series vs. non-consecutive.

Source

Table III Mol et al. 1998

glas *Tumor markers in the diagnosis of primary bladder cancer.*

Description

Outcome of individual studies evaluating urine markers

Format

A matrix with 46 rows and 7 columns. Each row represents study results, the columns are:

tp number of true positives.

n1 number of patients with disease.

fp number of false positives.

n2 number of patients without disease.

author first author of the study.

cutoff cutoff in U/ml.

marker test method used in the study.

Source

The data were obtained from

Glas AS, Roos D, Deutekom M, Zwindermann AH, Bossuyt PM, Kurth KH. (2003) Tumor markers in the diagnosis of primary bladder cancer. A systematic review. *Journal of Urology*; **169**:1975-82.

References

Verde P. E. (2010). Meta-analysis of diagnostic test data: A bivariate Bayesian modeling approach. *Statistics in Medicine*. **29**, 3088-3102.

`gould` *Accuracy of Positron Emission Tomography for Diagnosis of Pulmonary Nodules and Mass Lesions*

Description

Data from a Meta-Analysis of Studies Quality of FDG-PET for Diagnosis of SPNs and Mass Lesions

Format

A matrix with 31 rows and 6 columns. Each row represents study results, the columns are:

tp number of true positives.

n1 number of patients with disease.

fp number of false positives.

n2 number of patients without disease.

author first author of the study.

year publication date.

Source

The data were obtainded from

Gould MK, Maclean CC, Kuschner WG, Rydzak CE, Owens Dk. (2001) Accuracy of positron emission tomography for diagnosis of pulmonary nodules and mass lesions: a meta-analysis. *The Journal of the American Medical Association*;**285**:914-24.

`metadiag` *Bayesian Meta-Analysis of diagnostic test data*

Description

This function performers a Bayesian meta-analysis of diagnostic test data by fitting a bivariate random effects model. The number of true positives and false positives are modeled with two conditional Binomial distributions and the random-effects are based on a bivariate scale mixture of Normals. 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*.

Usage

```
metadiag(
  data,
  two.by.two = FALSE,
  re = "normal",
  re.model = "DS",
  link = "logit",
  mean.mu.D = 0,
  mean.mu.S = 0,
  sd.mu.D = 1,
  sd.mu.S = 1,
  sigma.D.upper = 10,
  sigma.S.upper = 10,
  mean.Fisher.rho = 0,
  sd.Fisher.rho = 1/sqrt(2),
  df = 4,
  df.estimate = FALSE,
  df.lower = 3,
  df.upper = 20,
  split.w = FALSE,
  n.1.new = 50,
  n.2.new = 50,
  nr.chains = 2,
  nr.iterations = 10000,
  nr.adapt = 1000,
  nr.burnin = 1000,
  nr.thin = 1,
  be.quiet = FALSE,
  r2jags = TRUE
)
```

Arguments

`data`	Either a data frame with at least 4 columns containing the true positives (tp), number of patients with disease (n1), false positives (fp), number of patients

	without disease (n2), or for two.by.two = TRUE a data frame where each line contains the diagnostic results as a two by two table, where the column names are: TP, FP, TN, FN.
`two.by.two`	If TRUE indicates that the diagnostic results are given as: TP, FP, TN, FN.
`re`	Random effects distribution for the resulting model. Possible values are *normal* for bivariate random effects and *sm* for scale mixtures.
`re.model`	If re.model = "DS" indicates that the sum and differences of TPR and FPR are modeled as random effects and re.model = "SeSp" indicates that the Sensitivity and Specificity are modeled as ranodm effects. The defualt value is re.model = "DS".
`link`	The link function used in the model. Possible values are *logit*, *cloglog probit*.
`mean.mu.D`	prior Mean of D, default value is 0.
`mean.mu.S`	prior Mean of S, default value is 0.
`sd.mu.D`	prior Standard deviation of D, default value is 1 (the prior of mu.D is a logistic distribution).
`sd.mu.S`	prior Standard deviation of S, default value is 1 (the prior of mu.S is a logistic distribution).
`sigma.D.upper`	Upper bound of the uniform prior of sigma.S, default value is 10.
`sigma.S.upper`	Upper bound of the uniform prior of sigma.S, default value is 10.
`mean.Fisher.rho`	Mean of rho in the Fisher scale default value is 0.
`sd.Fisher.rho`	Standard deviation of rho in the Fisher scale, default value is 1/sqrt(2).
`df`	If de.estimate = FALSE, then df is the degrees of freedom for the scale mixture distribution, default value is 4.
`df.estimate`	Estimate the posterior of df. The defualt value is FALSE.
`df.lower`	Lower bound of the prior of df. The defulat value is 3.
`df.upper`	Upper bound of the prior of df. The defulat value is 30.
`split.w`	Split the w parameter in two independent weights one for each random effect. The default value is FALSE.
`n.1.new`	Number of patients with disease in a predictive study default is 50.
`n.2.new`	Number of patients with non-disease in a predictive study default is 50.
`nr.chains`	Number of chains for the MCMC computations, default 5.
`nr.iterations`	Number of iterations after adapting the MCMC, default is 10000. Some models may need more iterations.
`nr.adapt`	Number of iterations in the adaptation process, defualt is 1000. Some models may need more iterations during adptation.
`nr.burnin`	Number of iteration discared for burnin period, default is 1000. Some models may need a longer burnin period.
`nr.thin`	Thinning rate, it must be a positive integer, the default value 1.
`be.quiet`	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.
`r2jags`	Which interface is used to link R to JAGS (rjags and R2jags) default value is R2Jags TRUE.

Details

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 metadiag. 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. (2010). Meta-analysis of diagnostic test data: A bivariate Bayesian modeling approach. Statistics in Medicine. 29(30):3088-102. doi: 10.1002/sim.4055.

Verde P. E. (2018). bamdit: An R Package for Bayesian Meta-Analysis of Diagnostic Test Data. Journal of Statisticsl Software. Volume 86, issue 10, pages 1–32.

Examples

```
## Not run:

# Example: data from Glas et al. (2003).....................................
library(bamdit)
data("glas")
glas.t <- glas[glas$marker == "Telomerase", 1:4]

glas.t <- glas[glas$marker == "Telomerase", 1:4]

# Simple visualization ...

plotdata(glas.t,                # Data frame
         two.by.two = FALSE     # Data is given as: (tp, n1, fp, n2)
         )

glas.m1 <- metadiag(glas.t,                # Data frame
                    two.by.two = FALSE,    # Data is given as: (tp, n1, fp, n2)
                    re = "normal",         # Random effects distribution
                    re.model = "DS",       # Random effects on D and S
                    link = "logit",        # Link function
                    sd.Fisher.rho   = 1.7, # Prior standard deviation of correlation
```

```
                    nr.burnin = 1000,      # Iterations for burnin
                    nr.iterations = 10000, # Total iterations
                    nr.chains = 2,         # Number of chains
                    r2jags = TRUE)         # Use r2jags as interface to jags


 summary(glas.m1, digit=3)

 plot(glas.m1,                    # Fitted model
      level = c(0.5, 0.75, 0.95), # Credibility levels
      parametric.smooth = TRUE)   # Parametric curve


# Plot results: based on a non-parametric smoother of the posterior predictive rates .......

plot(glas.m1,                    # Fitted model
     level = c(0.5, 0.75, 0.95), # Credibility levels
     parametric.smooth = FALSE)  # Non-parametric curve


# Using the pipe command in the package dplyr ..............................................

library(dplyr)

glas.t %>%
  metadiag(re = "normal", re.model ="SeSp") %>%
  plot(parametric.smooth = FALSE, color.pred.points = "red")



# Visualization of posteriors of hyper-parameters ......................................
library(ggplot2)
library(GGally)
library(R2jags)
attach.jags(glas.m1)
hyper.post <- data.frame(mu.D, mu.S, sigma.D, sigma.S, rho)
ggpairs(hyper.post,                  # Data frame
       title = "Hyper-Posteriors",          # title of the graph
       lower = list(continuous = "density") # contour plots
       )


#.....................................................................

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

#    2) Different parametrization of random effects in the link scale:
#         DS = "differences of TPR and FPR"
#         SeSp = "Sensitivity and Specificity"

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

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

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

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

#    7) 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.

# logit-normal-DS
m <- metadiag(glas.t, re = "normal", re.model = "DS", link = "logit")
summary(m)
plot(m)

# cloglog-normal-DS
summary(metadiag(glas.t, re = "normal", re.model = "DS", link = "cloglog"))

# probit-normal-DS
summary(metadiag(glas.t, re = "normal", re.model = "DS", link = "probit"))
# logit-normal-SeSp
summary(metadiag(glas.t, re = "normal", re.model = "SeSp", link = "logit"))

# cloglog-normal-SeSp
summary(metadiag(glas.t, re = "normal", re.model = "SeSp", link = "cloglog"))
# probit-normal-SeSp
summary(metadiag(glas.t, re = "normal", re.model = "SeSp", link = "probit"))

# logit-sm-DS
summary(metadiag(glas.t, re = "sm", re.model = "DS", link = "logit", df = 1))

# cloglog-sm-DS
summary(m<-metadiag(glas.t, re = "sm", re.model = "DS", link = "cloglog", df = 1))
plot(m, parametric.smooth = FALSE)

# probit-sm-DS
summary(m<-metadiag(glas.t, re = "sm", re.model = "DS", link = "probit", df = 1))
plot(m, parametric.smooth = FALSE)

# logit-sm-SeSp
summary(m<-metadiag(glas.t, re = "sm", re.model = "SeSp", link = "logit", df = 1))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))

# cloglog-sm-SeSp
summary(m<-metadiag(glas.t, re = "sm", re.model = "SeSp", link = "cloglog", df = 1))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))

# probit-sm-SeSp
summary(m<-metadiag(glas.t, re = "sm", re.model = "SeSp", link = "probit", df = 1))
```

```
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))

# logit-sm-DS-df
summary(m<-metadiag(glas.t, re = "sm", re.model = "DS", link = "logit",
 df.estimate = TRUE))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))

# cloglog-sm-DS-df
summary(m<-metadiag(glas.t, re = "sm", re.model = "DS", link = "cloglog",
df.estimate = TRUE))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))

# probit-sm-DS-df
summary(m<-metadiag(glas.t, re = "sm", re.model = "DS", link = "probit",
df.estimate = TRUE))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))

# logit-sm-SeSp-df
summary(m<-metadiag(glas.t, re = "sm", re.model = "SeSp", link = "probit",
df.estimate = TRUE))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))

# cloglog-sm-SeSp-df
summary(m<-metadiag(glas.t, re = "sm", re.model = "SeSp", link = "cloglog",
df.estimate = TRUE))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))

# probit-sm-SeSp-df
summary(m<-metadiag(glas.t, re = "sm", re.model = "SeSp", link = "probit",
df.estimate = TRUE))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))

# split.w ..............................................................

# logit-sm-DS
summary(m <- metadiag(glas.t, re = "sm", re.model = "DS", link = "logit", split.w = TRUE, df = 10))
plot(m)

# cloglog-sm-DS
summary(m<-metadiag(glas.t, re = "sm", re.model = "DS", link = "cloglog", split.w = TRUE, df = 4))
plot(m)

# probit-sm-DS
summary(m<-metadiag(glas.t, re = "sm", re.model = "DS", link = "probit", split.w = TRUE, df = 4))
plot(m, parametric.smooth = FALSE)

# logit-sm-SeSp
summary(m<-metadiag(glas.t, re = "sm", re.model = "SeSp", link = "logit", split.w = TRUE, df = 1))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))
plotw(m)

# cloglog-sm-SeSp
summary(m<-metadiag(glas.t, re = "sm", re.model = "SeSp", link = "cloglog", split.w = TRUE, df = 1))
```

```
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))
plotw(m)

# probit-sm-SeSp
summary(m<-metadiag(glas.t, re = "sm", re.model = "SeSp", link = "probit", split.w = TRUE, df = 1))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))
plotw(m)


# logit-sm-DS-df
summary(m<-metadiag(glas.t, re = "sm", re.model = "DS", link = "logit", split.w = TRUE,
 df.estimate = TRUE))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))
plotw(m)

# cloglog-sm-DS-df
summary(m<-metadiag(glas.t, re = "sm", re.model = "DS", link = "cloglog", split.w = TRUE,
df.estimate = TRUE))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))
plotw(m)

# probit-sm-DS-df
summary(m<-metadiag(glas.t, re = "sm", re.model = "DS", link = "probit", split.w = TRUE,
df.estimate = TRUE))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))
plotw(m)

# logit-sm-SeSp-df
summary(m<-metadiag(glas.t, re = "sm", re.model = "SeSp", link = "probit", split.w = TRUE,
df.estimate = TRUE))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))
plotw(m)

# cloglog-sm-SeSp-df
summary(m<-metadiag(glas.t, re = "sm", re.model = "SeSp", link = "cloglog", split.w = TRUE,
df.estimate = TRUE))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))
plotw(m)

# probit-sm-SeSp-df
summary(m<-metadiag(glas.t, re = "sm", re.model = "SeSp", link = "probit", split.w = TRUE,
df.estimate = TRUE))
plot(m, parametric.smooth = FALSE, level = c(0.5, 0.9))
plotw(m)


## End(Not run)
```

---

mri *Diagnosis of lymph node metastasis with magnetic resonance imaging*

Description

Diagnosis of lymph node metastasis with magnetic resonance imaging

Format

A matrix with 10 rows and 4 columns. Each row represents study results, the columns are:

tp true positives

n1 number of patients with disease

fp false positives

n2 number of patients without disease

Source

The data were obtained from

Scheidler J, Hricak H, Yu KK, Subak L, Segal MR. (1997) Radiological evaluation of lymph node metastases in patients with cervical cancer: a meta-analysis. *The Journal of the American Medical Association*; **278**:1096-1101.

References

Verde P. E. (2010). Meta-analysis of diagnostic test data: A bivariate Bayesian modeling approach. *Statistics in Medicine*. **29**, 3088-3102.

`plot.metadiag` *Generic plot function for metadiag object in bamdit*

Description

This function plots the observe data in the ROC (Receiving Operating Charachteristics) space with the posterior predictive contours. The predictive curves are approximated using a non-parametric smoother or with a parametric model. For the parametric model the current implementation supports only a logistic link function. The marginal posterior predictive distributions are ploted outside the ROC space.

Usage

```
## S3 method for class 'metadiag'
plot(
  x,
  parametric.smooth = TRUE,
  level = c(0.5, 0.75, 0.95),
  limits.x = c(0, 1),
  limits.y = c(0, 1),
  kde2d.n = 25,
  color.line = "red",
```

```
  title = paste("Posterior Predictive Contours (50%, 75% and 95%)"),
  marginals = TRUE,
  bin.hist = 30,
  color.hist = "lightblue",
  S = 500,
  color.pred.points = "lightblue",
  color.data.points = "blue",
  ...
)
```

Arguments

x	The object generated by the metadiag function.
parametric.smooth	Indicates if the predictive curve is a parametric or non-parametric.
level	Credibility levels of the predictive curve. If parametric.smooth = FALSE, then the probability levels are estimated from the nonparametric surface.
limits.x	Numeric vector of length 2 specifying the x-axis limits. The default value is c(0, 1).
limits.y	Numeric vector of length 2 specifying the x-axis limits. The default value is c(0, 1).
kde2d.n	The number of grid points in each direction for the non-parametric density estimation. Can be scalar or a length-2 inter vector.
color.line	Color of the predictive contour line.
title	Optional parameter for setting a title in the plot.
marginals	Plot the posterior marginal predictive histograms.
bin.hist	Number of bins of the marginal histograms.
color.hist	Color of the histograms.
S	Number of predictive rates to be plotted.
color.pred.points	Color of the posterior predictive rates.
color.data.points	Color of the data points.
...	...

See Also

metadiag.

Examples

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

```
data("glas")
glas.t <- glas[glas$marker == "Telomerase", 1:4]
glas.m1 <- metadiag(glas.t,                # Data frame
                    re = "normal",         # Random effects distribution
                    re.model = "DS",       # Random effects on D and S
                    link = "logit",        # Link function
                    sd.Fisher.rho   = 1.7, # Prior standard deviation of correlation
                    nr.burnin = 1000,      # Iterations for burnin
                    nr.iterations = 10000, # Total iterations
                    nr.chains = 2,         # Number of chains
                    r2jags = TRUE)         # Use r2jags as interface to jags


 plot(glas.m1,                    # Fitted model
      level = c(0.5, 0.75, 0.95), # Credibility levels
      parametric.smooth = TRUE)   # Parametric curve

# Plot results: based on a non-parametric smoother of the posterior predictive rates .......

plot(glas.m1,                    # Fitted model
     level = c(0.5, 0.75, 0.95), # Credibility levels
     parametric.smooth = FALSE)  # Non-parametric curve

# Using the pipe command in the package dplyr and changing some colors .......

library(dplyr)

glas.t %>%
 metadiag(re = "normal", re.model ="SeSp") %>%
   plot(parametric.smooth = FALSE,
          S = 100,
          color.data.points = "green",
          color.pred.points = "blue",
          color.line = "black")


## End(Not run)
```

---

plotcompare *plotcompare*

Description

This function compares the predictive posterior surfaces of two fitted models.

Usage

```
plotcompare(
  m1,
  m2,
```

```
  level = 0.95,
  title = paste("Comparative Predictive Posterior Contours"),
  m1.name = "Model.1",
  m2.name = "Model.2",
  group = NULL,
  limits.x = c(0, 1),
  limits.y = c(0, 1),
  group.colors = c("blue", "red")
)
```

Arguments

m1	A model fitted to the data. This is an object generated by the metadiag function.
m2	A second model fitted to the data. This is an object generated by the metadiag function.
level	Credibility level of the predictive curves.
title	The title of the plot.
m1.name	Label of the model 1.
m2.name	Label of the model 2.
group	An optional argument, which is a variable name indicating a group factor. This argument is used to compare results from two subgroups.
limits.x	A vector with the limits of the horizontal axis.
limits.y	A vector with the limits of the vertical axis.
group.colors	A character vector with two color names.

See Also

metadiag.

Examples

```
## execute analysis
## Not run:

# Comparing results from two models same data

data(glas)
glas.t <- glas[glas$marker == "Telomerase", 1:4]
glas.m1 <- metadiag(glas.t)
glas.m2 <- metadiag(glas.t, re = "sm")
plotcompare(m1 = glas.m1, m2 = glas.m2)

# Comparing results from two models fitted to two subgroups of data:
# studies with retrospective design and studies with prospective design

data("ct")
ct$design = factor(ct$design, labels = c("Prospective", "Retrospective"))
```

```
m1.ct <- metadiag(ct[ct$design=="Prospective", ])
m2.ct <- metadiag(ct[ct$design=="Retrospective", ])

plotcompare(m1.ct, m2.ct,m1.name = "Retrospective design",
m2.name = "Prospective design",group = "design",
limits.x = c(0, 0.75), limits.y = c(0.65, 1))



## End(Not run)
```

---

plotdata *Basic function to plot the data of meta-analysis of diagnostic test*

Description

This function plots the true positive rates vs the false positive rates of each study included in the meta-analysis. Study results are displayed by circles, the diameter of each circle is proportional to the sample size of the study (or table). If subgroups are displayed each group is represented by different colours. This function use the package *ggplot2*.

Usage

```
plotdata(
  data,
  two.by.two = FALSE,
  group = NULL,
  x.lo = 0,
  x.up = 1,
  y.lo = 0,
  y.up = 1,
  alpha.p = 0.7,
  max.size = 15
)
```

Arguments

data	Either a data frame with at least 4 columns containing the true positives (tp), number of patients with disease (n1), false positives (fp), number of patients without disease (n2), or for two.by.two = TRUE a data frame where each line contains the diagnostic results as a two by two table, where the column names are: TP, FP, TN, FN.
two.by.two	If TRUE indicates that the diagnostic results are given as: TP, FP, TN, FN.
group	a variable name indicating a group factor
x.lo	lower limit of the x-axis

x.up	upper limit of the x-axis
y.lo	lower limit of the y-axis
y.up	upper limit of the y-axis
alpha.p	transparency of the points
max.size	scale parameter of the maximum size

Examples

```
## execute analysis
## Not run:

data(ct)
ct$design <- with(ct, factor(design,
             labels = c("Prospective", "Retrospective")))

plotdata(ct,              # Data frame
        group = "design", # Groupping variable
        y.lo = 0.75,      # Lower limit of y-axis
        x.up = 0.75,      # Upper limit of x-axis
        alpha.p = 0.5,    # Transparency of the balls
        max.size = 5)     # Scale the circles

## End(Not run)
```

---

plotsesp *plotsesp() plot the posterior densities for Se and Sp*

Description

plotsesp() plot the posterior densities for Se and Sp

Usage

```
plotsesp(m, binwidth.p = 0.03, CI.level = 0.95)
```

Arguments

m	The object generated by the metadiag function.
binwidth.p	Histograms binwidth, default is 0.03.
CI.level	Level of the posterior interval default is 0.95.

See Also

metadiag.

Examples

```
## execute analysis
## Not run:
data(ep)
m1.ep <- metadiag(ep[,1:4])

plotsesp(m = m1.ep)

## End(Not run)
```

---

plotw *Plot for the conflict of evidence parameters w1 and w2*

Description

Conflict of evidence plot: this plot displays the posterior distribution of the study's weights w1 and w1. These weights indicate potential conflict of evidence of the studies. The weight w1 indicates deviations with respect to the specificity and w2 to the sensitivity.

Usage

```
plotw(
  m,
  group = NULL,
  title = "Posterior quantiles (25%, 50%, 75%)",
  group.colors = c("blue", "red")
)
```

Arguments

m	The object generated by metadiag. The model object must be fitted with the options: re = "sm" and split.w = TRUE.
group	An optional argument which is a variable name indicating a group factor. If set, then the plot is colored by groups.
title	The title of the plot.
group.colors	A character vector with two color names.

See Also

metadiag.

Examples

```
## execute analysis
## Not run:
data(ep)
ep$design = factor(ep$d1,labels = c("prospective", "retrospective"))
m.ep <- metadiag(ep, re = "sm", re.model = "SeSp",
                 split.w = TRUE,
                 df.estimate = TRUE)

plotw(m.ep)
#Relationship between conflict and study design
plotw(m.ep, group = "design")


## End(Not run)
```

`print.metadiag` *Generic print function for metadiag object in bamdit*

Description

Generic print function for metadiag object in bamdit

Usage

```
## S3 method for class 'metadiag'
print(x, digits = 3, ...)
```

Arguments

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

`rapt` *Systematic reviews of clinical decision tools for acute abdominal pain*

Description

This data frame corresponds to 13 clinical studies reporting the accuracy of doctors added with decision tools.

Format

A data frame with 13 rows and 13 columns. Each row represents study results, the columns are:

Author Name of the first author and year of publication

tp.dr Number of true positive cases for unadded doctors.

fp.dr Number of false positive cases for unadded doctors.

fn.dr Number of false negative cases for unadded doctors.

tn.dr Number of true negative cases for unadded doctors.

tp.tools Number of true positive cases for doctors with decision tools.

fp.tools Number of false positive cases for doctors with decision tools.

fn.tools Number of false negative cases for doctors with decision tools.

tn.tools Number of true negative cases for doctors with decision tools.

tool Diagnostic tool.

n.dr Total number of cases for unadded doctors.

n.tools Total number of cases for doctors with decision tools.

design Study design.

Details

This data frame contains results of diagnostic accuraccy of 13 studies which reported comparison of sensitivity and specificity between doctors using diagnostic tools vs doctors without decision tools.

Source

Health Technol Assess. 2006 Nov;10(47):1-167, iii-iv. Systematic reviews of clinical decision tools for acute abdominal pain. Liu JL1, Wyatt JC, Deeks JJ, Clamp S, Keen J, Verde P, Ohmann C, Wellwood J, Dawes M, Altman DG.

References

Health Technol Assess. 2006 Nov;10(47):1-167, iii-iv. Systematic reviews of clinical decision tools for acute abdominal pain. Liu JL1, Wyatt JC, Deeks JJ, Clamp S, Keen J, Verde P, Ohmann C, Wellwood J, Dawes M, Altman DG.

`safdar05` *Diagnosis of Intravascular Device-Related Bloodstream Infection*

Description

Outcome of individual studies evaluating intravascular device-related bloodstream infection

Format

A matrix with 78 rows and 8 columns. Each row represents study results, the columns are:

tp number of true positives.

n1 number of patients with disease.

fp number of false positives.

n2 number of patients without disease.

author first author of the study.

year publication date.

technique diagnostic technique used in the study.

duration duration of catheterization: short term or long term or both.

Source

The data were obtained from

Safdar N, Fine JP, Maki DG. (2005) Meta-analysis: methods for diagnosing intravascular device-related bloodstream infection. *Ann Intern Med.*; **142**:451-66.

`scheidler` *Radiological evaluation of lymph node metastases in patients with cervical cancer: a meta-analysis.*

Description

This data frame summarizes the tables 1-3 of Scheidler et al. 1997.

Format

A matrix with 46 rows and 7 columns. Each row represents study results, the columns are:

tp true positives.

n1 number of patients with disease.

fp false positives.

n2 number of patients without disease.

author first author of the study.

year publication date.

test test method used in the study.

Source

The data were obtained from

Scheidler J, Hricak H, Yu KK, Subak L, Segal MR. (1997) Radiological evaluation of lymph node metastases in patients with cervical cancer: a meta-analysis. *The Journal of the American Medical Association*; **278**:1096-1101.

References

Verde P. E. (2010). Meta-analysis of diagnostic test data: A bivariate Bayesian modeling approach. *Statistics in Medicine*. **29**, 3088-3102.

`skin` *Accuracy of Computer-Aided Diagnosis of Melanoma: A Meta-analysis.*

Description

This data frame contains results 70 studies investigated computer-aided diagnosis of melanoma

Format

A matrix with 70 rows and 15 columns. Each row represents a study's results, the columns are:

"TP" number of true positives.

"TN" number of ture negatives.

"FP" number of false positives.

"FN" number of false negative.

"study_ID" Study identification

"test_set_source" Public or propietary.

"method" Diagnostic technique used in the study: computer vision; deep learning or hardware-based.

"test_set_independent" yes or no.

"SAMPLE_SELECTION_BR" QUADAS-2, Patient selection bias.

"INDEX_TEST_BR" QUADAS-2, Index test description/application bias.

"REFERENCE_STANDARD_BR" QUADAS-2, Reference standard bias.

"FLOW_AND_TIMING_BR" QUADAS-2, Patient flow and timing bias.

"SAMPLE_SELECTION_AP QUADAS-2, Patient selection bias.

"INDEX_TEST_AP" QUADAS-2, Index test description/application bias.

"REFERENCE_STANDARD_AP" QUADAS-2, Reference standard bias.

Source

The data were obtained from

Dick V, Sinz C, Mittlböck M, Kittler H, Tschandl P. Accuracy of Computer-Aided Diagnosis of Melanoma: A Meta-analysis. JAMA Dermatol. 2019 Nov 1;155 11:1291-1299. doi: 10.1001/jamadermatol.2019.1375. PMID: 31215969; PMCID: PMC6584889.

summary.metadiag *Generic summary function for metadiag object in bamdit*

Description

Generic summary function for metadiag object in bamdit

Usage

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

Arguments

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

Index