Package ‘BayesCACE’

October 12, 2022

Type Package
Title Bayesian Model for CACE Analysis
Version 1.2.3
Date 2022-10-1
Depends R (>= 3.5.0), rjags (>= 4-6)
Imports coda, Rdpack, grDevices, forestplot, metafor, lme4, methods
SystemRequirements JAGS 4.x.y (http://mcmc-jags.sourceforge.net)
Description Performs CACE (Complier Average Causal Effect analysis) on either a single study or meta-analysis of datasets with binary outcomes, using either complete or incomplete noncompliance information. Our package implements the Bayesian methods proposed in Zhou et al. (2019) <doi:10.1111/biom.13028>, which introduces a Bayesian hierarchical model for estimating CACE in meta-analysis of clinical trials with noncompliance, and Zhou et al. (2021) <doi:10.1080/01621459.2021.1900859>, with an application example on Epidural Analgesia.
License GPL (>= 2)
NeedsCompilation no
RoxygenNote 7.1.2
RdMacros Rdpack
Encoding UTF-8
LazyData true
Suggests R.rsp
VignetteBuilder R.rsp
Author Jinhui Yang [aut, cre] (<https://orcid.org/0000-0001-8322-1121>), Jincheng Zhou [aut] (<https://orcid.org/0000-0003-2641-2495>), James Hodges [ctb], Haitao Chu [ctb] (<https://orcid.org/0000-0003-0932-598X>)
Maintainer Jinhui Yang <james.yangjinhui@gmail.com>
Repository CRAN
Date/Publication 2022-10-02 15:00:02 UTC
Bayesian hierarchical models for CACE meta-analysis with complete compliance data

This function performs the Bayesian hierarchical model method for meta-analysis when the dataset has complete compliance information for all studies, as described in Section 2.2, "the Bayesian hierarchical model", of the package manuscript.

Usage

cace.meta.c(
  data,
  param = c("CACE", "ulout", "vlout", "slout", "blout", "pic", "pin", "pia"),
  random.effects = list(),
  re.values = list(),
  model.code = "",
  digits = 3,
  n.adapt = 1000,
  n.iter = 1e+05,
  n.burnin = floor(n.iter/2),
  n.chains = 3,
  n.thin = max(1, floor((n.iter - n.burnin)/1e+05)),
  conv.diag = FALSE,
Arguments

data
an input dataset with the same structure as the example data epidural_c, containing multiple rows referring to multiple studies in a meta-analysis.

param
a character string vector indicating the parameters to be tracked and estimated. By default the following parameters (see details) are included: \( \theta^{\text{CACE}} \) (CACE), \( E(u_{11}) \) (u1out), \( E(v_{11}) \) (v1out), \( E(s_{11}) \) (s1out), \( \pi_a \) (pia), \( \pi_n \) (pin), and \( \pi_c = 1 - \pi_a - \pi_n \) (pic). Users can modify the string vector to only include parameters of interest besides \( \theta^{\text{CACE}} \).

random.effects
a list of logical values indicating whether random effects are included in the model. The list should contain the assignment for these parameters only: \( \delta_{in} \) (delta.n), \( \delta_{ia} \) (delta.a), \( \delta_{iu} \) (delta.u), \( \delta_{iv} \) (delta.v), \( \delta_{is} \) (delta.s), \( \delta_{ib} \) (delta.b), cor. The list should be in the form of list(delta.a = FALSE, cor = FALSE, ...). By default, this is an empty list, and all parameters are default to TRUE. Parameters that are not listed in the list are assumed to be TRUE. Note that \( \rho \) (cor) can only be included when both \( \delta_{in} \) (delta.n) and \( \delta_{ia} \) (delta.a) are set to TRUE. Otherwise, a warning occurs and the model continues running by forcing \( \delta_{in} = \text{TRUE} \) and \( \delta_{ia} = \text{TRUE} \).

re.values
a list of parameter values for the random effects. It should contain the assignment for these parameters only: \( \alpha_{nm} \) and \( \alpha_{ns} \), which refer to the mean and standard deviation used in the normal distribution estimation of \( \alpha_n \), as well as \( \alpha_{am} \), \( \alpha_{as} \), \( \alpha_{sm} \), \( \alpha_{ss} \), \( \alpha_{bm} \), \( \alpha_{bs} \), \( \alpha_{um} \), \( \alpha_{us} \), \( \alpha_{vm} \), \( \alpha_{vs} \). It also contains the shape and rate parameters of the gamma distributions of the standard deviation variable of \( \delta_{in} \), \( \delta_{ia} \), \( \delta_{iu} \), \( \delta_{iv} \), \( \delta_{is} \), \( \delta_{ib} \). The shape parameters are named as \( \tau_{n.h} \) and \( \tau_{a.h} \), for example, and the rate parameters are named as \( \tau_{n.r} \) and \( \tau_{a.r} \). You do not need to specify the shape and rate parameters if the corresponding random effect is set to FALSE in random.effects, since they will not be used anyways. By default, re.values is an empty list, and all the mean are set to 0, and \( \alpha_{ns} = \alpha_{as} = 0.16 \), and \( \alpha_{ss} = \alpha_{bs} = \alpha_{us} = \alpha_{vs} = 0.25 \), and the shape and rate parameters are default to 2.

model.code
a string representation of the model code; each line should be separated. Default to constructing model code using the model.meta.c function with the parameters that are inputted to this function. This parameter is only necessary if user wishes to make functional changes to the model code, such as changing the probability distributions of the parameters. Default to empty string.

digits
number of digits. Default to 3.

n.adapt
adapt value. Default to 1000.

n.iter
number of iterations. Default to 100000.

n.burnin
number of burn-in iterations. Default to n.iter/2.

n.chains
number of chains. Default to 3.
n.thin: thinning rate, must be a positive integer. Default to max(1, floor((n.iter - n.burnin)/100000)).

conv.diag: whether or not to show convergence diagnostics. Default to FALSE.

mcmc.samples: whether to include JAGS samples in the final output. Default to FALSE.

study.specific: a logical value indicating whether to calculate the study-specific $\theta_{i}^{\text{CACE}}$. If TRUE, the model will first check the logical status of arguments delta.u and delta.v. If both are FALSE, meaning that neither response rate $u_{i1}$ or $v_{i1}$ is modeled with a random effect, then the study-specific $\theta_{i}^{\text{CACE}}$ is the same across studies. The function gives a warning and continues by making study.specific = FALSE. Otherwise, the study-specific $\theta_{i}^{\text{CACE}}$ are estimated and saved as the parameter cacei.

Value

It returns a model object whose attribute type is cace.Bayes

References


See Also

cace.study, cace.meta.ic

Examples

data("epidural_c", package = "BayesCACE")
set.seed(123)
out.meta.c <- cace.meta.c(data = epidural_c, conv.diag = TRUE,
mcmc.samples = TRUE, study.specific = TRUE)
# By calling the object smry from the output list out.meta.c, posterior estimates
# (posterior mean, standard deviation, posterior median, 95\% credible interval, and
# time-series standard error) are displayed.
out.meta.c$smry
out.meta.c$DIC
Bayesian hierarchical models for CACE meta-analysis with incomplete compliance information

Description

This function also estimates $\theta^{CACE}$ using the Bayesian hierarchical model but can accommodate studies with incomplete compliance data. The necessary data structure and the likelihood function are presented in Section 2.3 of the package manuscript, "CACE for meta-analysis with incomplete compliance information".

Usage

cace.meta.ic(
  data,
  param = c("CACE", "u1out", "v1out", "s1out", "b1out", "pic", "pin", "pia"),
  random.effects = list(),
  re.values = list(),
  model.code = "",
  digits = 3,
  n.adapt = 1000,
  n.iter = 1e+05,
  n.burnin = floor(n.iter/2),
  n.chains = 3,
  n.thin = max(1, floor((n.iter - n.burnin)/1e+05)),
  conv.diag = FALSE,
  mcmc.samples = FALSE,
  study.specific = FALSE
)

Arguments

data an input dataset the same structure as the example data epidural_ic, containing multiple rows referring to multiple studies in a meta-analysis.

param the list of parameter used. Default to c("CACE", "u1out", "v1out", "s1out", "b1out", "pic", "pin", "pia").

random.effects a list of logical values indicating whether random effects are included in the model. The list should contain the assignment for these parameters only: delta.n ($\delta_{in}$), delta.a ($\delta_{ia}$), delta.u ($\delta_{iu}$), delta.v ($\delta_{iv}$), delta.s ($\delta_{is}$), delta.b ($\delta_{ib}$), cor. The list should be in the form of list(delta.a = FALSE, cor = FALSE, ...). By default, this is an empty list, and all parameters are default to TRUE. Parameters that are not listed in the list are assumed to be TRUE. Note that $\rho$ (cor) can only be included when both $\delta_{in}$ (delta.n) and $\delta_{ia}$ (delta.a) are set to TRUE. Otherwise, a warning occurs and the model continues running by forcing delta.n = TRUE and delta.a = TRUE.
re.values  a list of parameter values for the random effects. It should contain the assignment for these parameters only: alpha.n.m and alpha.n.s, which refer to the mean and standard deviation used in the normal distribution estimation of alpha.n, as well as alpha.a.m, alpha.a.s, alpha.s.m, alpha.s.s, alpha.b.m, alpha.b.s, alpha.u.m, alpha.u.s, alpha.v.m, alpha.v.s. It also contains the shape and rate parameters of the gamma distributions of the standard deviation variable of delta.n, delta.a, delta.u, delta.v, delta.s, delta.b. The shape parameters are named as tau.n.h and tau.a.h, for example, and the rate parameters are named as tau.n.r and tau.a.r. You do not need to specify the shape and rate parameters if the corresponding random effect is set to FALSE in random.effects, since they will not be used anyways. By default, re.values is an empty list, and all the mean are set to 0, and alpha.n.s = alpha.a.s = 0.16, and alpha.s.s = alpha.b.s = alpha.u.s = alpha.v.s = 0.25, and the shape and rate parameters are default to 2.

model.code  a string representation of the model code; each line should be separated. Default to constructing model code using the model.meta.ic function with the parameters that are inputted to this function. This parameter is only necessary if user wishes to make functional changes to the model code, such as changing the probability distributions of the parameters. Default to empty string.

digits  number of digits. Default to 3.

n.adapt  adapt value. Default to 1000.

n.iter  number of iterations. Default to 100000.

n.burnin  number of burn-in iterations. Default to n.iter/2.

n.chains  number of chains. Default to 3.

n.thin  thinning rate, must be a positive integer. Default to max(1,floor((n.iter-n.burnin)/100000)).

conv.diag  whether or not to show convergence diagnostics. Default to FALSE.

mcmc.samples  whether to include JAGS samples in the final output. Default to FALSE.

study.specific  a logical value indicating whether to calculate the study-specific $\theta^{CACE}_i$. If TRUE, the model will first check the logical status of arguments delta.u and delta.v. If both are FALSE, meaning that neither response rate $u_{i1}$ or $v_{i1}$ is modeled with a random effect, then the study-specific $\theta^{CACE}_i$ is the same across studies. The function gives a warning and continues by making study.specific = FALSE. Otherwise, the study-specific $\theta^{CACE}_i$ are estimated and saved as the parameter cacei.

**Details**

Note that when compiling the JAGS model, the warning ‘adaptation incomplete’ may occasionally occur, indicating that the number of iterations for the adaptation process is not sufficient. The default value of n.adapt (the number of iterations for adaptation) is 1,000. This is an initial sampling phase during which the samplers adapt their behavior to maximize their efficiency (e.g., a Metropolis–Hastings random walk algorithm may change its step size). The ‘adaptation incomplete’ warning indicates the MCMC algorithm may not achieve maximum efficiency, but it generally has little impact on the posterior estimates of the treatment effects. To avoid this warning, users may increase n.adapt.
Value

It returns a model object whose attribute type is cace.Bayes

References


See Also

cace.study, cace.meta.c

Examples

data("epidural_ic", package = "BayesCACE")
set.seed(123)
out.meta.ic <- cace.meta.ic(data = epidural_ic, conv.diag = TRUE,
mcmc.samples = TRUE, study.specific = TRUE)

cace.study

CACE analysis for a single study, or a two-step approach for meta-analysis with complete compliance information

Description

This function performs CACE analysis for a single study using the likelihood and model specified in Section 2.1 of the package manuscript, or a two-step approach for meta-analysis with complete compliance information as described in Section 2.2, “the two-step approach”.

Usage

cace.study(
data,
param = c("CACE", "u1", "v1", "s1", "b1", "pi.c", "pi.n", "pi.a"),
re.values = list(),
model.code = "",
digits = 3,
n.adapt = 1000,
n.iter = 1e+05,
n.burnin = floor(n.iter/2),
n.chains = 3,
n.thin = max(1, floor((n.iter - n.burnin)/1e+05)),
conv.diag = FALSE,
mcmc.samples = FALSE,
two.step = FALSE,
method = "REML"
)
Arguments

data a input dataset the same structure as the example data epidural_c, containing either one row of observations for a single study, or multiple rows referring to multiple studies in a meta-analysis.

param a character string vector indicating the parameters to be tracked and estimated. By default all parameters in the model (see details) are included: \( \theta^{CACE} \), \( u_1 \), \( v_1 \), \( s_1 \), \( b_1 \), \( \pi_a \), \( \pi_n \), and \( \pi_c = 1 - \pi_a - \pi_n \). Users can modify the string vector to only include parameters of interest besides \( \theta^{CACE} \).

re.values a list of parameter values for the random effects. It should contain the assignment for these parameters only: \( n.m \) and \( n.s \), which refer to the mean and standard deviation used in the normal distribution estimation of \( n \), as well as \( a.m \), \( a.s \), \( alpha.s.m \), \( alpha.s.s \), \( alpha.b.m \), \( alpha.b.s \), \( alpha.u.m \), \( alpha.u.s \), \( alpha.v.m \), and \( alpha.v.s \). By default, this is an empty list, and all the mean are set to 0, and \( alpha.n.s = alpha.a.s = 0.16 \) and \( alpha.s.s = alpha.b.s = alpha.u.s = alpha.v.s = 0.25 \).

model.code a string representation of the model code; each line should be separated. Default to constructing model code using the model.meta.ic function with the parameters that are inputted to this function. This parameter is only necessary if user wishes to make functional changes to the model code, such as changing the probability distributions of the parameters. Default to empty string.

digits a positive integer specifying the digits after the decimal point for the effect size estimates. The default is 3.

n.adapt the number of iterations for adaptation in Markov chain Monte Carlo (MCMC) algorithm; it is used to maximize the sampling efficiency. The default is 1,000. If a warning "adaptation incomplete" appears, users may increase n.adapt. This argument and the following n.iter, n.burnin, n.chains, n.thin are passed to the functions in R package rjags.

n.iter the number of iterations of each MCMC chain. The default is 100,000.

n.burnin the number of iterations for burn-in period. The default is the largest integer not greater than n.iter/2.

n.chains the number of MCMC chains. The default is 3.

n.thin a positive integer indicating thinning rate for MCMC chains, which is used to avoid potential high auto-correlation and to save computer memory when n.iter is large. The default is set as 1 or the largest integer not greater than \( (n.iter - n.burnin)/1e+05 \), whichever is larger.

conv.diag a logical value indicating whether to compute the Gelman and Rubin convergence statistic (\( R \)) of each parameter as a convergence diagnostic. It is considered the chains are well mixed and have converged to the target distribution if \( R \leq 1.1 \). The default is FALSE. If TRUE, n.chains must be greater than 1, and the function saves each chain's MCMC samples for all parameters, which can be used to produce trace, posterior density, and auto-correlation plots by calling the function plt.cacebayes.

mcmc.samples a logical value indicating whether to save MCMC posterior samples in the output object. The default is FALSE. If TRUE, the output object list includes each
chain's MCMC samples for all parameters. They can be used in the function plt.cacebayes to generate the trace, posterior density, and auto-correlation plots for further model diagnostics.

two.step

A logical value indicating whether to conduct a two-step meta-analysis. If two.step = TRUE, the posterior mean and standard deviation of study-specific $\theta_i^{CACE}$ are used to perform a standard meta-analysis, using the R package metafor.

method

The method used in meta-analysis if two.step = TRUE. The default estimation method is the REML (restricted maximum-likelihood estimator) method for the random-effects model. Users can change the argument method to obtain different meta-analysis estimators from either a random-effects model or a fixed-effect model, e.g., method = 'DL' refers to the DerSimonian–Laird estimator, method = 'HE' returns the Hedges estimator, and method = 'HS' gives the Hunter–Schmidt estimator. More details are available from the documentation of the function metafor::rma. If the input data include only one study, the meta-analysis result is just the same as the result from the single study.

Details

The likelihood

$$
\log L(\beta) = N_{000} \log(\pi_c(1 - v_1) + \pi_n(1 - s_1)) + N_{001} \log(\pi_c v_1 + \pi_n s_1) + N_{010} \log(\pi_a(1 - b_1)) \\
+ N_{011} \log(\pi_a b_1) + N_{100} \log(\pi_c(1 - s_1)) + N_{101} \log(\pi_n s_1) + N_{110} \log((\pi_c(1 - u_1) \\
+ \pi_a(1 - b_1)) + N_{111} \log(\pi_c u_1 + \pi_a b_1) + constant
$$

If the input data includes more than one study, the study-specific CACEs will be estimated by retrieving data row by row. By default, the function cace.study() returns a list including posterior estimates (posterior mean, standard deviation, median, and a 95% credible interval (CrI) with 2.5% and 97.5% quantiles as the lower and upper bounds), and the deviance information criterion (DIC) statistic for each study.

Value

It returns a model object whose attribute type is cace.Bayes

See Also

cace.meta.c, cace.meta.ic

Examples

data("epidural_c", package = "BayesCACE")
set.seed(123)
out.study <- cace.study(data = epidural_c, conv.diag = TRUE, mcmc.samples = TRUE, two.step = TRUE)
# Show the estimates of theta for each single study (posterior mean and
# standard deviation, posterior median, 95% credible interval, and time-series
# standard error):
out.study$CACE
# If the argument conv.diag is specified as TRUE, the output list contains
# a sub-list conv.out, which outputs the Gelman and Rubin convergence statistic,
# labelled Point est.) calculated for each parameter from each single study, and
# their upper confidence limits (labelled Upper C.I.).
out.study$conv.out[[1]]

coda.names   Get names of node array

Description
This is a helper function from the rjags library in order to get the names of the individual elements of a node array. See the package rjags for more details.

Usage

coda.names(basename, dim)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>basename</td>
<td>the node names</td>
</tr>
<tr>
<td>dim</td>
<td>dimension of the nodes</td>
</tr>
</tbody>
</table>

Value
It returns a list of the names of individual elements

References


coda.samples.dic Generate posterior samples in mcmc.list format

Description
This is a wrapper function for jags.samples which sets a trace monitor for all requested nodes, updates the model, and coerces the output to a single mcmc.list object. It also converts to the output to dic format. This function is based on the coda.samples function from the rjags library, and modified by Prof. Matthias Mittner.

Usage

coda.samples.dic(model, variable.names, n.iter, thin, ...)
**Arguments**

- `model`: a jags model object
- `variable.names`: a character vector giving the names of variables to be monitored
- `n.iter`: number of iterations to monitor
- `thin`: thinning interval for monitors
- `...`: optional arguments that are passed to the `jags.samples` method from the `rjags` library, for jags model objects

**Value**

It returns the output to the input model object, and in dic format.

**References**


[https://ihrke.github.io/post/2014/10/07/dicjags/](https://ihrke.github.io/post/2014/10/07/dicjags/)

**epidural_c**

*Meta-analysis data with full compliance information*

**Description**

The data contains a meta analysis of the association between using epidural analgesia in labor and the risk of cesarean section. It contains 10 trials with full compliance information, each with 8 observed counts.

**Usage**

```r
data(epidural_c)
```

**Format**

An object of class `data.frame` with 10 rows and 10 columns.

**Source**


**References**


**Examples**

```r
data(epidural_c)
```
model.meta.c

epidural_ic  Meta-analysis data without full compliance information

Description

The data contains a meta analysis of the association between using epidural analgesia in labor and the risk of cesarean section. It contains 27 studies, only 10 out of which have full compliance information.

Usage

data(epidural_ic)

Format

An object of class data.frame with 27 rows and 14 columns.

Source


References


Examples

data(epidural_ic)

model.meta.c  Bayesian hierarchical model code for CACE meta-analysis with complete compliance data

Description

This function generates part of the model code for meta-analysis when the dataset has complete compliance information for all studies, as described in Section 2.2, "the Bayesian hierarchical model" of the package manuscript. This function will be called internally if user uses the cace.meta.c function.

Usage

model.meta.c(random.effects = list(), re.values = list())
Arguments

random.effects a list of logical values indicating whether random effects are included in the model. The list should contain the assignment for these parameters only: delta.n (\(\delta_{in}\)), delta.a (\(\delta_{ia}\)), delta.u (\(\delta_{iu}\)), delta.v (\(\delta_{iv}\)), delta.s (\(\delta_{is}\)), delta.b (\(\delta_{ib}\)), cor. The list should be in the form of list(delta.a = FALSE, cor = FALSE, ...). By default, this is an empty list, and all parameters are default to TRUE. Parameters that are not listed in the list are assumed to be TRUE. Note that \(\rho\) (cor) can only be included when both \(\delta_{in}\) (delta.n) and \(\delta_{ia}\) (delta.a) are set to TRUE. Otherwise, a warning occurs and the model continues running by forcing delta.n = TRUE and delta.a = TRUE.

re.values a list of parameter values for the random effects. It should contain the assignment for these parameters only: alpha.n.m and alpha.n.s, which refer to the mean and standard deviation used in the normal distribution estimation of alpha.n, as well as alpha.a.m, alpha.a.s, alpha.s.m, alpha.s.s, alpha.b.m, alpha.b.s, alpha.u.m, alpha.u.s, alpha.v.m, alpha.v.s. It also contains the shape and rate parameters of the gamma distributions of the standard deviation variable of delta.n, delta.a, delta.u, delta.v, delta.s, delta.b. The shape parameters are named as tau.n.h and tau.a.h, for example, and the rate parameters are named as tau.n.r and tau.a.r. You do not need to specify the shape and rate parameters if the corresponding random effect is set to FALSE in random.effects, since they will not be used anyways. By default, re.values is an empty list, and all the mean are set to 0, and alpha.n.s = alpha.a.s = 0.16, and alpha.s.s = alpha.b.s = alpha.u.s = alpha.v.s = 0.25, and the shape and rate parameters are default to 2.

Value

It returns a model string

Examples

# use default settings
model.string <- model.meta.ic()

model.meta.ic Bayesian hierarchical model code for CACE meta-analysis with complete compliance data

Description

This function generates the model code for meta-analysis when the dataset has incomplete compliance information for all studies, as described in Section 2.2.2, “the Bayesian hierarchical model” of the package manuscript. This function will be called internally if user uses the cace.meta.ic function.

Usage

model.meta.ic(random.effects = list(), re.values = list())
Arguments

random.effects a list of logical values indicating whether random effects are included in the model. The list should contain the assignment for these parameters only: delta.n (δn), delta.a (δia), delta.u (δiu), delta.v (δiv), delta.s (δis), delta.b (δib), cor. The list should be in the form of list(delta.a = FALSE, cor = FALSE, ...). By default, this is an empty list, and all parameters are default to TRUE. Parameters that are not listed in the list are assumed to be TRUE. Note that ρ (cor) can only be included when both δn (delta.n) and δia (delta.a) are set to TRUE. Otherwise, a warning occurs and the model continues running by forcing delta.n = TRUE and delta.a = TRUE.

re.values a list of parameter values for the random effects. It should contain the assignment for these parameters only: alpha.n.m and alpha.n.s, which refer to the mean and standard deviation used in the normal distribution estimation of alpha.n, as well as alpha.a.m, alpha.a.s, alpha.s.m, alpha.s.s, alpha.b.m, alpha.b.s, alpha.u.m, alpha.u.s, alpha.v.m, alpha.v.s. It also contains the shape and rate parameters of the gamma distributions of the standard deviation variable of delta.n, delta.a, delta.u, delta.v, delta.s, delta.b. The shape parameters are named as tau.n.h and tau.a.h, for example, and the rate parameters are named as tau.n.r and tau.a.r. You do not need to specify the shape and rate parameters if the corresponding random effect is set to FALSE in random.effects, since they will not be used anyways. By default, re.values is an empty list, and all the mean are set to 0, and alpha.n.s = alpha.a.s = 0.16, and alpha.s.s = alpha.b.s = alpha.u.s = alpha.v.s = 0.25, and the shape and rate parameters are default to 2.

Value

It returns a model string

Examples

# use default settings
model.string <- model.meta.ic()

Description

This function generates the model code for a single study using the likelihood and model specified in Section 2.1, or a two-step approach for meta-analysis with complete compliance information as described in Section 2.2, "The two-step approach" of the package manuscript. This function will be called internally if user uses the cace.study function.

Usage

model.study(re.values = list())
Arguments

re.values  
a list of parameter values for the random effects. It should contain the assignment for these parameters only: n.m and n.s, which refer to the mean and standard deviation used in the normal distribution estimation of n, as well as a.m, a.s, alpha.s.m, alpha.s.s, alpha.b.m, alpha.b.s, alpha.u.m, alpha.u.s, alpha.v.m, alpha.v.s. By default, this is an empty list, and all the mean are set to 0, and alpha.n.s = alpha.a.s = 0.16, and alpha.s.s = alpha.b.s = alpha.u.s = alpha.v.s = 0.25.

Value

It returns a model string

Examples

model.string <- model.study()

Description

This is a helper function from the rjags library in order to parse the string of form "a" or "a[n,p;q,r]". See the package rjags for more details.

Usage

parse.varname(varname)

Arguments

varname  
string name of variable

Value

It returns a list of parsed parameters

References

plt.acf

**Description**

This function creates an acf (Autocorrelation Function) plot for a model object with the type attribute cace.Bayes.

**Usage**

plt.acf(obj, param = c("CACE"), trialnumber = 1, ...)

**Arguments**

- **obj**: a model object, returned by `cace.meta.c`, `cace.meta.ic`, or `cace.study`
- **param**: list of parameters to plot
- **trialnumber**: indicator for which trial number of the mcmc samples to use. The default is 1
- **...**: optional parameters to pass into the acf function from the stats library.

**Value**

It returns an acf plot in an R plot window.

**Examples**

```r
out.meta.c <- cace.meta.c(data = epidural_c, conv.diag = TRUE, mcmc.samples = TRUE, study.specfic = TRUE)
plt.acf(obj = out.meta.c)
```

---

plt.density

**Description**

This function creates a density plot for a model object with the type attribute cace.Bayes.

**Usage**

plt.density(obj, param = c("CACE"), trialnumber = 1, ...)

**Examples**

```r
out.meta.c <- cace.meta.c(data = epidural_c, conv.diag = TRUE, mcmc.samples = TRUE, study.specfic = TRUE)
plt.density(obj = out.meta.c)
```
plt.forest

Arguments

| obj         | a model object, returned by cace.meta.c, cace.meta.ic, or cace.study |
| param      | list of parameters to plot                                          |
| trialnumber | indicator for which trial number of the mcmc samples to use. The default is 1 |
| ...        | optional parameters to pass into the plot function                  |

Value

It returns a density plot in an R plot window.

Examples

```r
out.meta.c <- cace.meta.c(data = epidural_c, conv.diag = TRUE,
                           mcmc.samples = TRUE, study.specific = TRUE)
plt.density(obj=out.meta.c)
```

---

plt.forest

this plot function makes a forest plot.

---

Description

This function provides a visual overview (forest plot) for a model object and corresponding dataset.

Usage

```r
plt.forest(data, obj, ...)
```

Arguments

| data                          | an input dataset with the same structure as the example data epidural_c, containing multiple rows referring to multiple studies in a meta-analysis. |
| obj                           | a model object returned by cace.meta.c, cace.meta.ic, or cace.study |
| ...                           | optional parameters passed into the forestplot function from the forestplot library |

Value

It returns a forestplot object in an R plot window.

Examples

```r
data("epidural_c", package = "BayesCACE")
out.meta.c <- cace.meta.c(data = epidural_c, conv.diag = TRUE,
                           mcmc.samples = TRUE, study.specific = TRUE)
plt.forest(data=epidural_c, obj=out.meta.c)
```
plt.noncomp

Plotting noncompliance rates for a given dataset

Description

Provides a forest plot of noncompliance rates in an R plot window.

Usage

plt.noncomp(data, overall = TRUE, ...)

Arguments

data a dataset with structure like the example epidural_c or epidural_ic
overall a logical value indicating whether a summary estimate of the compliance rates per randomization group is provided. The default is TRUE. This overall rate is estimated using a logit generalized linear mixed model.
... optional parameters passed into the forestplot function from the forestplot library

Details

This function provides a visual overview (forest plot) of study-specific noncompliance rates in both randomization arms.

Only studies with full compliance information are included in this plot because noncompliance rates cannot be calculated without compliance data. In the generated plot, the red dot with its horizontal line shows the study-specific noncompliance rate with its 95% exact confidence interval for the patients randomized to the treatment arm, and the blue square with its horizontal line represents that rate and interval for those in the control arm. The confidence intervals are calculated by the Clopper–Pearson exact method, which is based on the cumulative distribution function of the binomial distribution.

Value

A forest plot of noncompliance rates in an R plot window

Examples

data("epidural_c", package = "BayesCACE")
plt.noncomp(data=epidural_c, overall = TRUE)
**plt.trace**

This plot function creates a traceplot

**Description**

This function creates a traceplot for a model object with the type attribute `cace.Bayes`.

**Usage**

```r
plt.trace(obj, param = c("CACE"), trialnumber = 1, ...)
```

**Arguments**

- `obj`: a model object, returned by `cace.meta.c`, `cace.meta.ic`, or `cace.study`
- `param`: list of parameters to plot
- `trialnumber`: indicator for which trial number of the mcmc samples to use. The default is 1
- `...`: optional parameters to pass into the `plot` function

**Value**

It returns a traceplot in an R plot window.

**Examples**

```r
out.meta.c <- cace.meta.c(data = epidural_c, conv.diag = TRUE, mcmc.samples = TRUE, study.specific = TRUE)
plt.trace(obj=out.meta.c)
```

**prior.meta**

The function returns a custom string that specifies part of the model.

**Description**

This function returns a partially complete prior string. Used internally - cannot be directly used.

**Usage**

```r
prior.meta(random.effects = list(), re.values = list())
```
Arguments

random.effects  a list of logical values indicating whether random effects are included in the model. The list should contain the assignment for these parameters only: delta.n (δn), delta.a (δa), delta.u (δu), delta.v (δv), delta.s (δs), delta.b (δb), cor. The list should be in the form of list(delta.a = FALSE, cor = FALSE, ...). By default, this is an empty list, and all parameters are default to TRUE. Parameters that are not listed in the list are assumed to be TRUE. Note that ρ (cor) can only be included when both δn (delta.n) and δa (delta.a) are set to TRUE. Otherwise, a warning occurs and the model continues running by forcing delta.n = TRUE and delta.a = TRUE.

re.values  a list of parameter values for the random effects. It should contain the assignment for these parameters only: alpha.n.m and alpha.n.s, which refer to the mean and standard deviation used in the normal distribution estimation of alpha.n, as well as alpha.a.m, alpha.a.s, alpha.s.m, alpha.s.s, alpha.b.m, alpha.b.s, alpha.u.m, alpha.u.s, alpha.v.m, alpha.v.s. It also contains the shape and rate parameters of the gamma distributions of the standard deviation variable of delta.n, delta.a, delta.u, delta.v, delta.s, delta.b. The shape parameters are named as tau.n.h and tau.a.h, for example, and the rate parameters are named as tau.n.r and tau.a.r. You do not need to specify the shape and rate parameters if the corresponding random effect is set to FALSE in random.effects, since they will not be used anyways. By default, re.values is an empty list, and all the mean are set to 0, and alpha.n.s = alpha.a.s = 0.16, and alpha.s.s = alpha.b.s = alpha.u.s = alpha.v.s = 0.25, and the shape and rate parameters are default to 2.

Value

custom prior string

Examples

model.string <- prior.meta()

prior.study

The function returns a custom string that specifies part of the model (single-study).

Description

This function returns a partially complete prior string. Used internally - cannot be directly used.

Usage

prior.study(re.values = list())
Arguments

re.values  a list of parameter values for the random effects. It should contain the assignment for these parameters only: n.m and n.s, which refer to the mean and standard deviation used in the normal distribution estimation of n, as well as a.m, a.s, alpha.s.m, alpha.s.s, alpha.b.m, alpha.b.s, alpha.u.m, alpha.u.s, alpha.v.m, alpha.v.s. By default, this is an empty list, and all the mean are set to 0, and alpha.n.s = alpha.a.s = 0.16, and alpha.s.s = alpha.b.s = alpha.u.s = alpha.v.s = 0.25.

Value

custom model string

Examples

model.string <- prior.study()
Index

* dataset
  epidural_c, 11
  epidural_ic, 12

cace.meta.c, 2, 7, 9
ecase.meta.ic, 4, 5, 9
cace.study, 4, 7, 7
coda.names, 10
coda.samples.dic, 10

epidural_c, 11
epidural_ic, 12

model.meta.c, 12
model.meta.ic, 13
model.study, 14

parse.varname, 15
plt.acf, 16
plt.density, 16
plt.forest, 17
plt.noncomp, 18
plt.trace, 19
prior.meta, 19
prior.study, 20