Package ‘RprobitB’

November 12, 2021

Type Package

Title Bayes Estimation of Latent Class Mixed Multinomial Probit Models

Version 1.0.0

Date 2021-11-11

Description Fitting latent class mixed multinomial probit (LCMMNP) models to simulated or empirical choice data via Bayesian estimation. The number of latent classes can be updated within the algorithm on a weight-based strategy. For a reference on the method see Oelschläger and Bauer (2021) <https://trid.trb.org/view/1759753>.

License GPL-3

Encoding UTF-8

Imports Rcpp, mvtnorm, viridis

LinkingTo Rcpp, RcppArmadillo

Suggests knitr, rmarkdown, mlogit, vdiffr, testthat (>= 3.0.0)

Depends R (>= 3.5.0)

RoxygenNote 7.1.2

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Author Lennart Oelschläger [aut, cre], Dietmar Bauer [aut], Sebastian Büscher [ctb], Manuel Batram [ctb]

Maintainer Lennart Oelschläger <lennart.oelschlaeger@uni-bielefeld.de>

Repository CRAN

Date/Publication 2021-11-12 16:50:09 UTC
R topics documented:

- choice_probs
- classify
- compare
- compute_log_likelihood
- delta
- dmvnrm_arma_mc
- is_covariance_matrix
- mcmc
- overview_effects
- plot.RprobitB_model
- predict
- prepare
- print.RprobitB_data
- print.RprobitB_gibbs_samples_statistics
- print.RprobitB_latent_classes
- print.RprobitB_model
- print.RprobitB_normalization
- print.summary.RprobitB_data
- print.summary.RprobitB_model
- rdirichlet
- RprobitB_data
- RprobitB_gibbs_samples_statistics
- RprobitB_latent_classes
- RprobitB_model
- RprobitB_normalization
- RprobitB_normalization
- RprobitB_parameter
- rwishart
- R_hat
- set_mfrow
- simulate
- summary.RprobitB_data
- summary.RprobitB_model
- transform
- undiff_Sigma

Index

choice_probs  Compute choice probabilities of an RprobitB_model.

Description

This function computes the choice probabilities of an RprobitB_model.
**Usage**

```r
classify(x)
```

**Arguments**

- `x`: An object of class `RprobitB_model`.

**Value**

A data frame with the deciders id and the latent class number.

---

**Description**

This function classifies deciders based on the estimated latent classes.

- **Usage**
  ```r
classify(x)
  ```

- **Arguments**
  - `x`: An object of class `RprobitB_model`.

- **Value**
  - A data frame with the deciders id and the latent class number.

---

**compare**

*Compare fitted models.*

**Description**

This function computes model comparison criteria.

**Usage**

```r
compare(...)  
```

**Arguments**

- `...`: One or more objects of class `RprobitB_model`.
Value

A matrix with each model’s

• number of parameters,
• log-likelihood,
• AIC, and
• BIC value.

---

compute_log_likelihood

*Compute log-likelihood of an RprobitB_model.*

Description

This function computes the log-likelihood of an RprobitB_model.

Usage

compute_log_likelihood(x, at_true = FALSE)

Arguments

x
An object of class RprobitB_model.

at_true
If TRUE, choice probabilities are computed at the true parameters.

Value

The log-likelihood value.

---

delta

*Difference operator.*

Description

This function computes the difference operator matrix for computing utility differences.

Usage

delta(J, i)

Arguments

J
The total number of alternatives.

i
The alternative number to which respect utility differences are computed.
Details

Given a $J \times P$ matrix $X$ of choice characteristics, then $\delta(i, J) \times X$ computes differences with respect to alternative $i$.

Value

A matrix of dimension $(J-1) \times J$.

---

**dmvnrm arma mc**

*Multivariate normal density*

**Description**

Function to compute the density of a multivariate normal distribution.

**Usage**

```r
dmvnrm_arma_mc(x, mean, sigma, logd = FALSE)
```

**Arguments**

- `x` A matrix, the arguments.
- `mean` A vector, the mean.
- `sigma` A matrix, the covariance matrix.
- `logd` A boolean, whether to apply the logarithm.

**Value**

A vector, the computed multivariate normal densities

---

**is covariance matrix**

*Check covariance matrix properties.*

**Description**

This function checks if the input is a proper covariance matrix, i.e. a symmetric, numeric matrix with non-negative eigenvalues.

**Usage**

```r
is_covariance_matrix(x)
```

**Arguments**

- `x` A matrix.
mcmc

Perform Markov chain Monte Carlo simulation for fitting a (latent class) (mixed) (multinomial) probit model.

Description

This function performs Markov chain Monte Carlo simulation for fitting a (latent class) (mixed) (multinomial) probit model to discrete choice data.

Usage

mcmc(
  data,
  scale = list(parameter = "s", index = 1, value = 1),
  R = 10000,
  B = R/2,
  Q = 1,
  print_progress = TRUE,
  prior = NULL,
  latent_classes = NULL,
  seed = NULL
)

Arguments

data An object of class RprobitB_data.
scale A named list of three elements, determining the parameter normalization with respect to the utility scale:
  - parameter: Either "a" (for a linear coefficient of "alpha") or "s" (for a variance of the error-term covariance matrix "Sigma").
  - index: The index of the parameter that gets fixed.
  - value: The value for the fixed parameter.
R The number of iterations of the Gibbs sampler.
B The length of the burn-in period, i.e. a non-negative number of samples to be discarded.
Q The thinning factor for the Gibbs samples, i.e. only every Qth sample is kept.
print_progress A boolean, determining whether to print the Gibbs sampler progress and the estimated remaining computation time.
prior A named list of parameters for the prior distributions of the normalized parameters:
  - eta: The mean vector of length P_f of the normal prior for alpha.
• **Psi**: The covariance matrix of dimension \( P_f \times P_f \) of the normal prior for \( \alpha \).
• **delta**: The concentration parameter of length 1 of the Dirichlet prior for \( s \).
• **xi**: The mean vector of length \( P_r \) of the normal prior for each \( b_c \).
• **D**: The covariance matrix of dimension \( P_r \times P_r \) of the normal prior for each \( b_c \).
• **nu**: The degrees of freedom (a natural number greater than \( P_r \)) of the Inverse Wishart prior for each \( \Omega_c \).
• **Theta**: The scale matrix of dimension \( P_r \times P_r \) of the Inverse Wishart prior for each \( \Omega_c \).
• **kappa**: The degrees of freedom (a natural number greater than \( J-1 \)) of the Inverse Wishart prior for \( \Sigma \).
• **E**: The scale matrix of dimension \( J-1 \times J-1 \) of the Inverse Wishart prior for \( \Sigma \).

**latent_classes**

Either **NULL** or a list of parameters specifying the number and the latent classes:

• **C**: The number (greater or equal 1) of latent classes, which is set to 1 per default and is ignored if \( P_r = 0 \). If \( \text{update} = \text{TRUE} \), \( C \) equals the initial number of classes.
• **update**: A boolean, determining whether to update \( C \). Ignored if \( P_r = 0 \). If \( \text{update} = \text{FALSE} \), all of the following elements are ignored.
• **Cmax**: The maximum number of latent classes.
• **buffer**: The updating buffer (number of iterations to wait before the next update).
• **epsmin**: The threshold weight for removing latent classes (between 0 and 1).
• **epsmax**: The threshold weight for splitting latent classes (between 0 and 1).
• **distmin**: The threshold difference in means for joining latent classes (non-negative).

**seed**

Set a seed for the Gibbs sampling.

**Details**

See the vignette "Model fitting" for more details: vignette("model_fitting",package = "RprobitB").

**Value**

An object of class `RprobitB_model`.

**Examples**

```r
# Not run:
## probit model
p = simulate(form = choice ~ var | 0, N = 100, T = 10, J = 2, seed = 1)
m1 = mcmc(data = p, seed = 1)

## multinomial probit model
mnp = simulate(form = choice ~ var | 0, N = 100, T = 10, J = 3, seed = 1)
```
overview_effects

### mixed multinomial probit model
mmnp = simulate(form = choice ~ 0 | var, N = 100, T = 10, J = 3, re = "var", seed = 1)
m3 = mcmc(data = mmnp, seed = 1)

### mixed multinomial probit model with 2 latent classes
lcmmnp = simulate(form = choice ~ 0 | var, N = 100, T = 10, J = 3, re = "var", seed = 1, C = 2)
m4 = mcmc(data = lcmmnp, latent_classes = list("C" = 2), seed = 1)

### update of latent classes
m5 = mcmc(data = lcmmnp, latent_classes = list("update" = TRUE), seed = 1)

## End(Not run)

overview_effects  Overview of effects.

**Description**

This function gives an overview of the linear coefficients and whether they are connected to random effects.

**Usage**

overview_effects(form, re = NULL, alternatives)

**Arguments**

form  A formula object that is used to specify the probit model. The structure is choice ~ A | B | C, where

- A are names of alternative and choice situation specific covariates with a generic coefficient,
- B are names of choice situation specific covariates with alternative specific coefficients,
- and C are names of alternative and choice situation specific covariates with alternative specific coefficients.

Separate multiple covariates of one type by a + sign. By default, alternative specific constants (ASCs) are added to the model (for all except for the last alternative). They can be removed by adding +0 in the second spot. See the vignette vignette("data_management", package = "RprobitB") for more details.

re  A character (vector) of covariates of form with random effects. If re = NULL (the default), there are no random effects. To have random effects for the alternative specific constants, include "ASC" in re.

alternatives  A character vector with the names of the choice alternatives. If not specified, the choice set is defined by the observed choices.
**plot.RprobitB_model**

**Value**

A data frame with the coefficient names and booleans indicating whether they are connected to random effects.

**Description**

This function is the plot method for an object of class `RprobitB_model`.

**Usage**

```r
## S3 method for class 'RprobitB_model'
plot(x, type = "effects", ignore = NULL, ...)
```

**Arguments**

- `x`: An object of class `RprobitB_model`.
- `type`: The type of plot, which can be one or more of:
  - "effects" (the default) for visualizing the linear effects.
  - "mixture" for visualizing the mixture distribution,
  - "acf" for autocorrelation plots of the Gibbs samples,
  - "trace" for trace plots of the Gibbs samples.
- `ignore`: A character (vector) of covariate or parameter names that do not get visualized.
- `...`: Ignored.

**Value**

No return value. Draws a plot to the current device.

**predict**

*Predict choices.*

**Description**

This function predicts the choices of decision makers.

**Usage**

```r
predict(x, data = NULL, overview = TRUE)
```
Arguments

**x**  An object of class `RprobitB_model`.

**data**  Either `NULL` or an object of class `RprobitB_data`.

**overview**  If `TRUE`, aggregate the prediction in a table.

Value

Either a table if `overview = TRUE` or a data frame otherwise.

Description

This function prepares empirical choice data for the RprobitB package.

Usage

```r
prepare(
  form,
  choice_data,
  alternatives = NULL,
  re = NULL,
  id = "id",
  standardize = NULL,
  test_prop = NULL
)
```

Arguments

**form**  A formula object that is used to specify the probit model. The structure is `choice ~ A | B | C`, where

- A are names of alternative and choice situation specific covariates with a generic coefficient,
- B are names of choice situation specific covariates with alternative specific coefficients,
- and C are names of alternative and choice situation specific covariates with alternative specific coefficients.

Separate multiple covariates of one type by a `+` sign. By default, alternative specific constants (ASCs) are added to the model (for all except for the last alternative). They can be removed by adding `+0` in the second spot. See the vignette `vignette("data_management", package = "RprobitB")` for more details.

**choice_data**  A data frame of choice data in "wide" format (i.e. each row represents one choice occasion) with the following requirements:
• It must contain a column named id which contains unique identifier for each decision maker.
• It can contain a column named choice with the observed choices, where choice must match the name of the dependent variable in form. Such a column is required for model fitting but not for prediction.
• For each alternative specific covariate \( p \) in form and each choice alternative \( j \) in alternatives, choice_data must contain a column named \( p_j \).
• For each covariate \( q \) in form that is constant across alternatives, choice_data must contain a column named \( q \).

alternatives A character vector with the names of the choice alternatives. If not specified, the choice set is defined by the observed choices.
re A character (vector) of covariates of form with random effects. If \( re = \text{NULL} \) (the default), there are no random effects. To have random effects for the alternative specific constants, include "ASC" in re.
id A character, the name of the column in choice_data that contains unique identifier for each decision maker. The default is "id".
standardize A character vector of names of covariates that get standardized. Covariates of type 1 or 3 have to be addressed by \(<\text{covariate}>_<\text{alternative}>\). If standardize = "all", all covariates get standardized.
test_prop Either \( \text{NULL} \) or a numeric between 0 and 1. In the latter case, the data is split into a training set (of decider proportion \( 1 - \text{test\_prop} \)) and a testing set (of decider proportion \( \text{test\_prop} \)).

Details

See the vignette "Data management" for more details: vignette("data_management", package = "RprobitB").

Value

An object of class RprobitB_data. If test_prop is specified, a list of two RprobitB_data objects labelled "train" and "test".

Examples

data("Train", package = "mlogit")
data = prepare(form = choice ~ price | 0 | time + comfort + change, choice_data = Train, re = c("price","time"), standardize = "all")
print.RprobitB_data  

Print method for RprobitB_data.

Description

This function is the print method for an object of class RprobitB_data.

Usage

## S3 method for class 'RprobitB_data'
print(x, ...)

Arguments

x       An object of class RprobitB_data.
...     Ignored.

print.RprobitB_gibbs_samples_statistics

Print method for RprobitB_gibbs_samples_statistics.

Description

This function is the print method for an object of class RprobitB_gibbs_samples_statistics.

Usage

## S3 method for class 'RprobitB_gibbs_samples_statistics'
print(x, true = NULL, digits = 2, ...)

Arguments

x       An object of class RprobitB_gibbs_samples_statistics.
true    Either NULL or an object of class RprobitB_true_parameter.
digits  The number of printed decimal places.
...     Ignored.
print.RprobitB_latent_classes

Print method for RprobitB_latent_classes.

Description
This function is the print method for an object of class RprobitB_latent_classes.

Usage
## S3 method for class 'RprobitB_latent_classes'
print(x, ...)

Arguments
x                An object of class RprobitB_latent_classes.
...             Ignored.

print.RprobitB_model  Print method for RprobitB_model.

Description
This function is the print method for an object of class RprobitB_model.

Usage
## S3 method for class 'RprobitB_model'
print(x, ...)

Arguments
x                An object of class RprobitB_model.
...             Ignored.
**print.RprobitB_normalization**

*Print method for RprobitB_normalization.*

**Description**

This function is the print method for an object of class `RprobitB_normalization`.

**Usage**

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

**Arguments**

- `x` An object of class `RprobitB_normalization`.
- `...` Ignored.

---

**print.summary.RprobitB_data**

*Print method for the summary of RprobitB_data.*

**Description**

This function is the print method for an object of class `summary.RprobitB_data`.

**Usage**

```r
## S3 method for class 'summary.RprobitB_data'
print(x, ...)
```

**Arguments**

- `x` An object of class `summary.RprobitB_data`.
- `...` Ignored.
print.summary.RprobitB_model

Print method for the summary of RprobitB_model.

Description
This function is the print method for an object of class summary.RprobitB_model.

Usage
## S3 method for class 'summary.RprobitB_model'
print(x, digits = 2, ...)  

Arguments
x An object of class summary.RprobitB_model.
digits The number of printed decimal places.
... Ignored.

rdirichlet Draw from Dirichlet

Description
Function to draw from a Dirichlet distribution.

Usage
rdirichlet(alpha)

Arguments
alpha A vector, the concentration parameter.

Value
A vector, the sample from the Dirichlet distribution.
RprobitB_data

Create object of class RprobitB_data.

Description

This function creates an object of class RprobitB_data.

Usage

RprobitB_data(
  data,
  choice_data,
  N,
  T,
  J,
  P_f,
  P_r,
  alternatives,
  form,
  re,
  ASC,
  linear_coeffs,
  standardize,
  simulated,
  choice_available,
  true_parameter
)

Arguments

data A list with the choice data. The list has N elements. Each element is a list with two elements, \(X\) and \(y\), which are the covariates and decisions for a decision maker. More precisely, \(X\) is a list of \(T\) elements, where each element is a matrix of dimension \(J \times (P_f + P_r)\) and contains the characteristics for one choice occasion. \(y\) is a vector of length \(T\) and contains the labels for the chosen alternatives.

choice_data A data frame of choice data in "wide" format (i.e. each row represents one choice occasion) with the following requirements:

- It must contain a column named \(id\) which contains unique identifier for each decision maker.
- It can contain a column named choice with the observed choices, where choice must match the name of the dependent variable in \(form\). Such a column is required for model fitting but not for prediction.
- For each alternative specific covariate \(p\) in \(form\) and each choice alternative \(j\) in \(alternatives\), \(choice\_data\) must contain a column named \(p_j\).
- For each covariate \(q\) in \(form\) that is constant across alternatives, \(choice\_data\) must contain a column named \(q\).
**RprobitB_data**

N  The number (greater or equal 1) of decision makers.
T  The number (greater or equal 1) of choice occasions or a vector of choice occasions of length N (i.e. a decision maker specific number).
J  The number (greater or equal 2) of choice alternatives.
P_f The number of covariates connected to a fixed coefficient (can be 0).
P_r The number of covariates connected to a random coefficient (can be 0).
alternatives A character vector with the names of the choice alternatives. If not specified, the choice set is defined by the observed choices.
form A formula object that is used to specify the probit model. The structure is choice ~ A | B | C, where

- A are names of alternative and choice situation specific covariates with a generic coefficient,
- B are names of choice situation specific covariates with alternative specific coefficients,
- and C are names of alternative and choice situation specific covariates with alternative specific coefficients.

Separate multiple covariates of one type by a + sign. By default, alternative specific constants (ASCs) are added to the model (for all except for the last alternative). They can be removed by adding +0 in the second spot. See the vignette vignette("data_management",package = "RprobitB") for more details.
re A character (vector) of covariates of form with random effects. If re = NULL (the default), there are no random effects. To have random effects for the alternative specific constants, include "ASC" in re.
ASC A boolean, determining whether the model has ASCs.
linear_coeffs A data frame with the coefficient names and booleans indicating whether they are connected to random effects.
standardize A character vector of names of covariates that get standardized. Covariates of type 1 or 3 have to be addressed by <covariate>_<alternative>. If standardize = "all", all covariates get standardized.
simulated A boolean, if TRUE then data is simulated, otherwise data is empirical.
choice_available A boolean, if TRUE then data contains observed choices.
true_parameter An object of class RprobitB_parameters.

**Value**

An object of class RprobitB_data with the arguments of this function as elements.
RprobitB_gibbs_samples_statistics

*Compute parameter statistics.*

**Description**

This function computes parameter statistics from Gibbs samples.

**Usage**

```r
RprobitB_gibbs_samples_statistics(gibbs_samples, FUN)
```

**Arguments**

- `gibbs_samples` An object of class `RprobitB_gibbs_samples`.
- `FUN` A (preferably named) list of functions that compute parameter statistics from the Gibbs samples, i.e.
  - `mean` for the mean,
  - `sd` for the standard deviation,
  - `min` for the minimum,
  - `max` for the maximum,
  - `median` for the median,
  - `function(x) quantile(x,p)` for the pth quantile,
  - `R_hat` for the Gelman-Rubin statistic.

**Value**

An object of class `RprobitB_gibbs_samples_statistics`, which is a list of statistics from `gibbs_samples` obtained by applying the elements of `FUN`.

RprobitB_latent_classes

*Create object of class* `RprobitB_latent_classes`.

**Description**

This function creates an object of class `RprobitB_latent_classes`.

**Usage**

```r
RprobitB_latent_classes(latent_classes = NULL)
```
Arguments

latent_classes Either NULL or a list of parameters specifying the number and the latent classes:

• C: The number (greater or equal 1) of latent classes, which is set to 1 per default and is ignored if \( P_r = 0 \). If \( \text{update} = \text{TRUE} \), C equals the initial number of classes.

• update: A boolean, determining whether to update C. Ignored if \( P_r = 0 \). If \( \text{update} = \text{FALSE} \), all of the following elements are ignored.

• Cmax: The maximum number of latent classes.

• buffer: The updating buffer (number of iterations to wait before the next update).

• epsmin: The threshold weight for removing latent classes (between 0 and 1).

• epsmax: The threshold weight for splitting latent classes (between 0 and 1).

• distmin: The threshold difference in means for joining latent classes (non-negative).

Value

An object of class \texttt{RprobitB\_latent\_classes}.

\begin{center}
\texttt{RprobitB\_model(}
\hspace{1cm}
\texttt{Create object of class RprobitB\_model.)}
\end{center}

Description

This function creates an object of class \texttt{RprobitB\_model}.

Usage

\begin{verbatim}
RprobitB_model(
data,
normalization,
R,
B,
Q,
latent_classes,
prior,
gibbs_samples,
classification)
\end{verbatim}
Arguments

- data: An object of class RprobitB_data.
- normalization: An object of class RprobitB_normalization.
- R: The number of iterations of the Gibbs sampler.
- B: The length of the burn-in period, i.e. a non-negative number of samples to be discarded.
- Q: The thinning factor for the Gibbs samples, i.e. only every Qth sample is kept.
- latent_classes: Either NULL or a list of parameters specifying the number and the latent classes:
  - C: The number (greater or equal 1) of latent classes, which is set to 1 per default and is ignored if P_r = 0. If update = TRUE, C equals the initial number of classes.
  - update: A boolean, determining whether to update C. Ignored if P_r = 0. If update = FALSE, all of the following elements are ignored.
  - Cmax: The maximum number of latent classes.
  - buffer: The updating buffer (number of iterations to wait before the next update).
  - epsmin: The threshold weight for removing latent classes (between 0 and 1).
  - epsmax: The threshold weight for splitting latent classes (between 0 and 1).
  - distmin: The threshold difference in means for joining latent classes (non-negative).
- prior: A named list of parameters for the prior distributions of the normalized parameters:
  - eta: The mean vector of length P_f of the normal prior for alpha.
  - Psi: The covariance matrix of dimension P_f x P_f of the normal prior for alpha.
  - delta: The concentration parameter of length 1 of the Dirichlet prior for s.
  - xi: The mean vector of length P_r of the normal prior for each b_c.
  - D: The covariance matrix of dimension P_r x P_r of the normal prior for each b_c.
  - nu: The degrees of freedom (a natural number greater than P_r) of the Inverse Wishart prior for each Omega_c.
  - Theta: The scale matrix of dimension P_r x P_r of the Inverse Wishart prior for each Omega_c.
  - kappa: The degrees of freedom (a natural number greater than J-1) of the Inverse Wishart prior for Sigma.
  - E: The scale matrix of dimension J-1 x J-1 of the Inverse Wishart prior for Sigma.
- gibbs_samples: An object of class RprobitB_gibbs_samples.
- classification: The allocation variable of the estimated latent classes.

Value

An object of class RprobitB_model, i.e. a list with the arguments of this function as elements.
RprobitB_normalization

Create object of class RprobitB_normalization.

Description

This function creates an object of class RprobitB_normalization.

Usage

RprobitB_normalization(
  J,
  P_f,
  level = J,
  scale = list(parameter = "s", index = 1, value = 1)
)

Arguments

J          The number (greater or equal 2) of choice alternatives.
P_f        The number of covariates connected to a fixed coefficient (can be 0).
level      The number of the alternative with respect which utility differences are computed. Currently, only level = J (i.e. utility differences with respect to the last alternative) is implemented.
scale      A named list of three elements, determining the parameter normalization with respect to the utility scale:
            • parameter: Either "a" (for a linear coefficient of "alpha") or "s" (for a variance of the error-term covariance matrix "Sigma").
            • index: The index of the parameter that gets fixed.
            • value: The value for the fixed parameter.

Details

Any choice model has to be normalized with respect to level and scale.

• For level normalization, we takes utility differences with respect to one alternative.
• For scale normalization, we fix a model parameter. Per default, the first error-term variance is fixed to 1, i.e. scale = list("parameter" = "s", "index" = 1, "value" = 1). Alternatively, any error-term variance or any linear coefficient can be fixed.

Value

An object of class RprobitB_normalization, which is a list of the elements level and scale.
RprobitB_parameter  

Create object of class RprobitB_parameter.

Description

This function creates an object of class RprobitB_parameter. If sample = TRUE, missing parameters are sampled. All parameters are checked against the values of P_f, P_r, J, and N.

Usage

RprobitB_parameter(
  P_f,
  P_r,
  J,
  N,
  alpha = NULL,
  C = NULL,
  s = NULL,
  b = NULL,
  Omega = NULL,
  Sigma = NULL,
  Sigma_full = NULL,
  beta = NULL,
  z = NULL,
  seed = NULL,
  sample = TRUE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_f</td>
<td>The number of covariates connected to a fixed coefficient (can be 0).</td>
</tr>
<tr>
<td>P_r</td>
<td>The number of covariates connected to a random coefficient (can be 0).</td>
</tr>
<tr>
<td>J</td>
<td>The number (greater or equal 2) of choice alternatives.</td>
</tr>
<tr>
<td>N</td>
<td>The number (greater or equal 1) of decision makers.</td>
</tr>
<tr>
<td>alpha</td>
<td>The fixed coefficient vector of length P_f. Set to NA if P_f = 0.</td>
</tr>
<tr>
<td>C</td>
<td>The number (greater or equal 1) of latent classes of decision makers. Set to NA if P_r = 0. Otherwise, C = 1 per default.</td>
</tr>
<tr>
<td>s</td>
<td>The vector of class weights of length C. Set to NA if P_r = 0. For identifiability, the vector must be non-ascending.</td>
</tr>
<tr>
<td>b</td>
<td>The matrix of class means as columns of dimension P_r x C. Set to NA if P_r = 0.</td>
</tr>
<tr>
<td>Omega</td>
<td>The matrix of class covariance matrices as columns of dimension P_r*P_r x C. Set to NA if P_r = 0.</td>
</tr>
<tr>
<td>Sigma</td>
<td>The differenced error term covariance matrix of dimension J-1 x J-1 with respect to alternative J.</td>
</tr>
</tbody>
</table>
**Sigma_full**

The error term covariance matrix of dimension J x J. Internally, Sigma_full gets differenced with respect to alternative J, so it becomes an identified covariance matrix of dimension J-1 x J-1. If Sigma is specified, Sigma_full is ignored.

**beta**

The matrix of the decision-maker specific coefficient vectors of dimension P_r x N. Set to NA if P_r = 0.

**z**

The vector of the allocation variables of length N. Set to NA if P_r = 0.

**seed**

Set a seed for sampling missing parameters.

**sample**

A boolean, if TRUE missing parameters get sampled.

---

### Value

An object of class RprobitB_parameter, i.e. a named list with the model parameters alpha, C, s, b, Omega, Sigma, Sigma_full, beta, and z.

---

### rwishart

*Draw from a Wishart*

**Description**

Function to draw from Wishart and inverted Wishart distribution.

**Usage**

`rwishart(nu, V)`

**Arguments**

- **nu**
  
  A double, the degrees of freedom.

- **V**
  
  A matrix, the scale matrix.

**Value**

A list, the draw from the Wishart (W), inverted Wishart (IW), and corresponding Cholesky decomposition (C and CI)
**R_hat**  
*Compute Gelman-Rubin statistic.*

**Description**  
This function computes the Gelman-Rubin statistic $R_{hat}$.

**Usage**  
`R_hat(samples, parts = 2)`

**Arguments**

- `samples`: A vector or a matrix of samples from a Markov chain, e.g. Gibbs samples. If `samples` is a matrix, each column gives the samples for a separate run.
- `parts`: The number of parts to divide each chain into sub-chains.

**Value**

The Gelman-Rubin statistic.

**References**

https://bookdown.org/rdpeng/advstatcomp/monitoring-convergence.html

**Examples**

```r
no_chains = 2
do_chains = 1e3
samples = matrix(NA, length_chains, no_chains)
samples[1,] = 1
Gamma = matrix(c(0.8, 0.1, 0.2, 0.9), 2, 2)
for(c in 1:no_chains) for(t in 2:length_chains)
  samples[t,c] = sample(1:2, 1, prob = Gamma[samples[t-1,c],])
R_hat(samples)
```

---

**set_mfrow**  
*Balancing visualization of multiple figures.*

**Description**  
This function finds a balanced setting for `par(mfrow)`.

**Usage**  
`set_mfrow(n)`
Arguments

n  The total number of figures.

Value

A vector of the form c(nr, nc). If par(mfrow = c(nr, nc)), subsequent figures will be drawn in an
nr x nc array on the current device by rows.

simulate

Simulate choice data.

Description

This function simulates choice data for the RprobitB package.

Usage

simulate(
  form,
  N,
  T,
  J,
  re = NULL,
  alternatives = NULL,
  distr = NULL,
  standardize = NULL,
  seed = NULL,
  test_prop = NULL,
  ...
)

Arguments

form  A formula object that is used to specify the probit model. The structure is
      choice ~ A | B | C, where
      • A are names of alternative and choice situation specific covariates with a
generic coefficient,
      • B are names of choice situation specific covariates with alternative specific
coefficients,
      • and C are names of alternative and choice situation specific covariates with
alternative specific coefficients.
      Separate multiple covariates of one type by a + sign. By default, alternative spe-
cific constants (ASCs) are added to the model (for all except for the last alter-
native). They can be removed by adding +0 in the second spot. See the vignette
vignette("data_management",package = "RprobitB") for more details.

N  The number (greater or equal 1) of decision makers.
The number (greater or equal 1) of choice occasions or a vector of choice occasions of length N (i.e. a decision maker specific number).

The number (greater or equal 2) of choice alternatives.

A character (vector) of covariates of form with random effects. If re = NULL (the default), there are no random effects. To have random effects for the alternative specific constants, include "ASC" in re.

A character vector with the names of the choice alternatives. If not specified, the choice set is defined by the observed choices.

A named list of number generation functions from which the covariates are drawn. Covariates for which no distribution is specified are drawn from a standard normal distribution. Each element of distr must be of the form "cov" = list("name" = "<name of the number generation function>",...), where cov is the name of the covariate and ... are required parameters for the number generation function. Possible number generation functions are

- functions of the type r* from base R (e.g. rnorm) where all required parameters (except for n) must be specified,
- the function sample, where all required parameters (except for size) must be specified.

A character vector of names of covariates that get standardized. Covariates of type 1 or 3 have to be addressed by <covariate>_<alternative>. If standardize = "all", all covariates get standardized.

Set a seed for the simulation.

Either NULL or a numeric between 0 and 1. In the latter case, the data is split into a training set (of decider proportion 1-test_prop) and a testing set (of decider proportion test_prop).

Optionally specify alpha, C, s, b, Omega, Sigma, Sigma_full, beta, or z for the simulation.

See the vignette "Data management" for more details: vignette("data_management", package = "RprobitB").

An object of class RprobitB_data. If test_prop is specified, a list of two RprobitB_data objects labelled "train" and "test".

data = simulate(form = choice ~ cost | income + 0 | time, N = 100, T = 10, J = 3, re = "cost", alternatives = c("car", "bus", "scooter"))
summary.RprobitB_data  Summary method for RprobitB_data.

Description
This function is the summary method for an object of class RprobitB_data.

Usage
## S3 method for class 'RprobitB_data'
summary(object, ...)

Arguments
object       An object of class RprobitB_data.
...          Ignored.

summary.RprobitB_model  Summary method for RprobitB_model.

Description
This function is the summary method for an object of class RprobitB_model.

Usage
## S3 method for class 'RprobitB_model'
summary(object, FUN = c(mean = mean, sd = sd, `R^` = R_hat), ...)

Arguments
object       An object of class RprobitB_model.
FUN          A (preferably named) list of functions that compute parameter statistics from the
             Gibbs samples, i.e.
             • mean for the mean,
             • sd for the standard deviation,
             • min for the minimum,
             • max for the maximum,
             • median for the median,
             • function(x) quantile(x,p) for the pth quantile,
             • R_hat for the Gelman-Rubin statistic.
...          Ignored.
transform

Change the length of the burn-in period, the thinning factor and the scale after Gibbs sampling.

Description

Given an object of class `RprobitB_model`, this function can:

- change the length \( B \) of the burn-in period,
- change the the thinning factor \( Q \) of the Gibbs samples,
- change the model scale.

Usage

```r
transform(x, B = NULL, Q = NULL, scale = NULL, check_preference_flip = TRUE)
```

Arguments

- **x**: An object of class `RprobitB_model`.
- **B**: The length of the burn-in period, i.e. a non-negative number of samples to be discarded.
- **Q**: The thinning factor for the Gibbs samples, i.e. only every \( Q \)th sample is kept.
- **scale**: A named list of three elements, determining the parameter normalization with respect to the utility scale:
  - **parameter**: Either "a" (for a linear coefficient of "alpha") or "s" (for a variance of the error-term covariance matrix "Sigma").
  - **index**: The index of the parameter that gets fixed.
  - **value**: The value for the fixed parameter.
- **check_preference_flip**: If `TRUE` check for flip in preferences with new scale.

Details

See the vignette "Model fitting" for more details: `vignette("model_fitting",package = "RprobitB")`.

Value

An object of class `RprobitB_model`.
undiff_Sigma

Transform differenced to non-differenced error term covariance matrix.

Description

This function transforms the differenced error term covariance matrix $\Sigma$ back to a non-differenced error term covariance matrix.

Usage

undiff_Sigma($\Sigma$, $i$)

Arguments

- $\Sigma$  An error term covariance matrix of dimension $(J-1) \times (J-1)$ which was differenced with respect to alternative $i$.
- $i$  An integer, the alternative number with respect to which $\Sigma$ was differenced.

Value

A covariance matrix of dimension $J \times J$. If this covariance matrix gets differenced with respect to alternative $i$, the results is again $\Sigma$. 

Index

* helper
  delta, 4
  is_covariance_matrix, 5
  R_hat, 24
  set_mfrow, 24
  undiff_Sigma, 29

* s3
  RprobitB_data, 16
  RprobitB_gibbs_samples_statistics, 18
  RprobitB_latent_classes, 18
  RprobitB_model, 19
  RprobitB_normalization, 21
  RprobitB_parameter, 22
  rwishart, 23

choice_probs, 2
classify, 3
compare, 3
compute_log_likelihood, 4

delta, 4
dmvnrn arma_mc, 5

is_covariance_matrix, 5
mcmc, 6

overview_effects, 8

plot.RprobitB_model, 9
predict, 9
prepare, 10
print.RprobitB_data, 12
print.RprobitB_gibbs_samples_statistics, 12
print.RprobitB_latent_classes, 13
print.RprobitB_model, 13
print.RprobitB_normalization, 14
print.summary.RprobitB_data, 14
print.summary.RprobitB_model, 15

R_hat, 24
rdirichlet, 15
RprobitB_data, 16
RprobitB_gibbs_samples_statistics, 18
RprobitB_latent_classes, 18
RprobitB_model, 9, 19, 28
RprobitB_normalization, 21
RprobitB_parameter, 22

set_mfrow, 24
simulate, 25
summary.RprobitB_data, 27
summary.RprobitB_model, 27

transform, 28

undiff_Sigma, 29