Package ‘rmsBMA’

March 14, 2026

Title Reduced Model Space Bayesian Model Averaging

Version 0.1.2

Description Implements Bayesian model averaging for settings with many candidate regressors relative to the available sample size, including cases where the number of regressors exceeds the number of observations. By restricting attention to models with at most M regressors, the package supports reduced model space inference, thereby preserving degrees of freedom for estimation. It provides posterior summaries, Extreme Bounds Analysis, model selection procedures, joint inclusion measures, and graphical tools for exploring model probabilities, model size distributions, and coefficient distributions. The methodological approach follows Doppelhofer and Weeks (2009) <doi:10.1002/jae.1046>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Imports ggplot2, ggpubr, grid, gridExtra, Matrix, stats, tidyr, utils

Depends R (>= 3.5.0)

LazyData true

Language en-US

NeedsCompilation no

Author Krzysztof Beck [aut, cre] (ORCID: <https://orcid.org/0000-0003-3679-2962>)

Maintainer Krzysztof Beck <beckkrzysztof@gmail.com>

Repository CRAN

Date/Publication 2026-03-14 16:20:02 UTC

Contents

`best_models` *Table with the best models according to one of the posterior criteria*

Description

This function creates a ranking of best models according to one of the possible criterion (PMP under binomial model prior, PMP under binomial-beta model prior, R^2 under binomial model prior, R^2 under binomial-beta model prior). The function gives two types of tables in three different formats: inclusion table (where 1 indicates presence of the regressor in the model and 0 indicates that the variable is excluded from the model) and estimation results table (it displays the best models and estimation output for those models: point estimates, standard errors, significance level, and R^2).

Usage

```
best_models(bma_list, criterion = 1, best = 5, round = 3, estimate = TRUE)
```

Arguments

bma_list	bma object (the result of the bma function)
criterion	The criterion that will be used for a basis of the model ranking: 1 - binomial model prior 2 - binomial-beta model prior
best	The number of the best models to be considered
round	Parameter indicating the decimal place to which number in the tables should be rounded (default round = 3)
estimate	A parameter with values TRUE or FALSE indicating which table should be displayed when TRUE - table with estimation to the results FALSE - table with the inclusion of regressors in the best models

Value

A list with best_models objects:

1. matrix with inclusion of the regressors in the best models
2. matrix with estimation output in the best models with regular standard errors
3. knitr_kable table with inclusion of the regressors in the best models (the best for the display on the console - up to 11 models)
4. knitr_kable table with estimation output in the best models with regular standard errors (the best for the display on the console - up to 6 models)
5. gTree table with inclusion of the regressors in the best models (displayed as a plot). Use grid::grid.draw() to display.
6. gTree table with estimation output in the best models with regular standard errors (displayed as a plot). Use grid::grid.draw() to display.

Examples

```
data <- Trade_data[,1:10]
modelSpace <- model_space(data, M = 9, g = "UIP")
bma_list <- bma(modelSpace)
models <- best_models(bma_list, best = 3)
models[[4]]
```

bma — *Calculation of the bma objects*

Description

This function calculates bma and related objects for the modelSpace object obtained using model_space function.

Usage

```
bma(
  modelSpace,
  EMS = NULL,
  dilution = 0,
  dil.Par = 0.5,
  Narrative = 0,
  p = NULL,
  Nar_vec = NULL,
  round = 6
)
```

Arguments

`modelSpace`	Model space object (the result of the model_space function)
`EMS`	Expected model size for model binomial and binomial-beta model prior.
`dilution`	Binary parameter: 0 - NO application of a dilution prior; 1 - application of a dilution prior (George 2010).
`dil.Par`	Parameter associated with dilution prior - the exponent of the determinant (George 2010). Used only if parameter dilution=1.
`Narrative`	Binary parameter: 0 - NO application of a Narrative dilution prior; 1 - application of a Narrative dilution prior.
`p`	Parameter or vector that indicates by how much we cut probability of a model with substitutes.
`Nar_vec`	Vector with information on narrative dilution prior where: 0 - the variable has no substitutes; numbers different than 0 denote consecutive groups of variables considered to be substitutes.
`round`	Parameter indicating to which place the function should round up the results in final tables.

Value

A list with Posterior objects:

1. pmp_uniform_table - table with results with PMP under binomial model prior

2. pmp_random_table - table with results with PMP under binomial-beta model prior
3. eba_object - table with results of Extreme Bounds Analysis
4. pms_table - table with prior and posterior model sizes
5. x_names - vector with names of the regressors - to be used by the functions
6. K - total number of regressors
7. MS - size of the mode space
8. EMS - expected model size for binomial and binomial-beta model prior specified by the user (default EMS=K/2)
9. dilution - parameter indicating use of dilution
10. for_jointness - table for jointness function
11. for_best_models - table for best_models function
12. for_model_pmp - table for model_pmp function
13. for_model_sizes - table for model_sizes function
14. alphas - vector with the values of the constant
15. betas_nonzero - matrix with coefficients on regressors

Examples

```
data("Trade_data", package = "rmsBMA")
data <- Trade_data
modelSpace <- model_space(data, M = 6)
bma_list <- bma(modelSpace)
bma_list[[1]]
```

`coef_hist` *Graphs of the distribution of the coefficients over the model space*

Description

This function draws graphs of the distribution (in the form of histogram or kernel density) of the coefficients for all the considered regressors over the part of the model space that includes this regressors (half of the model space).

Arguments

`bma_list` bma object (the result of the bma function)

`weight` Parameter indicating whether the coefficients should be weighted by posterior model probabilities:

1. NULL - no weighting (default option)
2. "binomial" - using posterior model probabilities based on binomial model prior
3. "beta" - using posterior model probabilities based on binomial-beta model prior

`BW` Parameter indicating what method should be chosen to find bin widths for the histograms:

1. "FD" Freedman-Diaconis method
2. "SC" Scott method
3. "vec" user specified bin widths provided through a vector (parameter: binW)

`binW` A vector with bin widths to be used to construct histograms for the regressors. The vector must be of the size equal to total number of regressors. The vector with bin widths is used only if parameter BW="vec".

`BN` Parameter taking the values (default: BN = 0):
1 - the histogram will be build based on the number of bins specified by the user through parameter num. If BN=1, the function ignores parameters BW.
0 - the histogram will be build in line with parameter BW

`num` A vector with the numbers of bins used to be used to construct histograms for the regressors. The vector must be of the size equal to total number of regressors. The vector with bin widths is used only if parameter BN=1.

`kernel` A parameter taking the values (default: kernel = 0):
1 - the function will build graphs using kernel density for the distribution of coefficients (with kernel=1, the function ignores parameters BW and BN)
0 - the function will build regular histogram density for the distribution of coefficients

Value

A list with the graphs of the distribution of coefficients for all the considered regressors.

Examples

```
data("Trade_data", package = "rmsBMA")
data <- Trade_data[,1:10]
modelSpace <- model_space(data, M = 9, g = "UIP")
bma_list <- bma(modelSpace)
coefs <- coef_hist(bma_list, kernel = 1)
coefs[[1]]
coefs[[2]]
```

---

coef_to_full *Extracting coefficients from g_regression and fast_ols function*

Description

The function extracts coefficients or standard errors from the results of g_regression and fast_ols function

Usage

```
coef_to_full(model_coefs, model_row)
```

Arguments

`model_coefs`	a vector with estimated coefficients or standard errors
`model_row`	inclusion vector - row of a model space matrix

Value

A vector of coefficients coefficients or standard errors from the results of g_regression and fast_ols function `data_prep` *Introduction of time and section fixed effects and data standardization.*

Description

If the data is in the panel form the function assumes it has the following structure

section_1 year_1 y x1 x2 x3 ....
section_2 year_1 y x1 x2 x3 ....
section_3 year_1 y x1 x2 x3 ....
....
section_n year_1 y x1 x2 x3 ....
section_1 year_2 y x1 x2 x3 ....
section_2 year_2 y x1 x2 x3 ....
section_3 year_2 y x1 x2 x3 ....
....
section_n year_2 y x1 x2 x3 ....
....
section_n year_(T-1) y x1 x2 x3 ....
section_1 year_T y x1 x2 x3 ....
section_2 year_T y x1 x2 x3 ....
section_3 year_T y x1 x2 x3 ....
....
section_n year_T y x1 x2 x3 ....

Usage

```
data_prep(
  data,
  FE = FALSE,
  Time = 0,
  Section = 0,
  Time_FE = 0,
  Section_FE = 0,
  STD = 0
)
```

Arguments

`data`	A data file.
`FE`	Binary variable: TRUE - include fixed effect, FALSE - do not include fixed effects.
`Time`	The number of time periods - works only if FE=1.
`Section`	The number of cross-sections - works only if EF=1.

`Time_FE` Binary variable: 1 - include time fixed effect, 0 - do not include time fixed effects. Works only if EF=1.

`Section_FE` Binary variable: 1 - include cross-section fixed effect, 0 - do not include cross-section fixed effects. Works only if EF=1.

`STD` Binary variable: 1 - standardize the data set, 0 - do not standardize the data set. By standardization we mean subtraction of a mean and division by standard deviation of each variable.

Value

Formatted data set.

Examples

```
y <- matrix(1:20,nrow=20,ncol=1)
x1 <- matrix(21:40,nrow=20,ncol=1)
x2 <- matrix(41:60,nrow=20,ncol=1)
data <- cbind(y,x1,x2)
new_data <- data_prep(data,FE=1,Time=5,Section=4,Time_FE=1,Section_FE=1,STD=0)

y <- rnorm(20, mean = 0, sd = 1)
x1 <- rnorm(20, mean = 0, sd = 1)
x2 <- rnorm(20, mean = 0, sd = 1)
data <- cbind(y,x1,x2)
new_data <- data_prep(data,FE=1,Time=5,Section=4,Time_FE=1,Section_FE=1,STD=1)
```

`data_preparation` *Fixed-effects demeaning and data standardization*

Description

Prepares a dataset for econometric analysis by applying fixed-effects demeaning (within transformation) and/or standardization to numeric variables. The behavior of the function depends on whether panel identifiers are supplied and whether fixed effects are explicitly requested.

Usage

```
data_preparation(
  data,
  id = NULL,
  time = NULL,
  fixed_effects = FALSE,
  effect = c("twoway", "section", "time"),
  standardize = FALSE
)
```

Arguments

- `data` — A data.frame containing the data.
- `id` — An optional character string specifying the cross-sectional (section) identifier. Must be supplied together with `time` to enable fixed-effects demeaning.
- `time` — An optional character string specifying the time identifier. Must be supplied together with `id` to enable fixed-effects demeaning.
- `fixed_effects` — Logical. If `TRUE`, fixed-effects demeaning is applied when both `id` and `time` are provided. If `FALSE`, fixed-effects demeaning is skipped even when identifiers are present.
- `effect` — A character string indicating the fixed-effects structure when `fixed_effects = TRUE`. One of `"twoway"`, `"section"`, or `"time"`.
- `standardize` — Logical. If `TRUE`, numeric variables are standardized by subtracting their mean and dividing by their standard deviation. When fixed effects are applied, standardization occurs after demeaning.

Details

If both `id` and `time` are provided and `fixed_effects = TRUE`, the function applies section, time, or two-way fixed-effects demeaning and may optionally standardize the transformed variables. If `fixed_effects = FALSE`, fixed-effects demeaning is skipped even when identifiers are present, and only standardization (if requested) is applied.

If either `id` or `time` is missing, fixed-effects demeaning is not available and the function requires `standardize = TRUE`.

For two-way fixed effects, the transformation is:

$$x^*_{it} = x_{it} - \bar{x}_{i\cdot} - \bar{x}_{\cdot t} + \bar{x}_{\cdot\cdot}$$

Standardization consists of subtracting the mean and dividing by the standard deviation of each variable and is applied after fixed-effects demeaning (if any).

The function operates in three modes:

- **Fixed effects only**: `fixed_effects = TRUE`, `standardize = FALSE`.
- **Fixed effects + standardization**: `fixed_effects = TRUE`, `standardize = TRUE`.
- **Standardization only**: `fixed_effects = FALSE`, `standardize = TRUE`.

When `id` and `time` are not provided, only the standardization-only mode is available.

Missing values are ignored when computing means and standard deviations. After fixed-effects demeaning, an intercept term is redundant in subsequent linear regressions.

Value

A data.frame containing only numeric variables used in estimation. Panel identifiers (`id`, `time`) are removed from the output. Transformed variables preserve their original column names.

Examples

```
df <- migration_panel
# Standardization only (panel identifiers present but FE skipped)
X <- data_preparation(
  df,
  id = "Pair_ID",
  time = "Year_0",
  fixed_effects = FALSE,
  standardize = TRUE
)

# Two-way fixed effects with standardization
X <- data_preparation(
  df,
  id = "Pair_ID",
  time = "Year_0",
  fixed_effects = TRUE,
  effect = "twoway",
  standardize = TRUE
)

# Section fixed effects only
X <- data_preparation(
  df,
  id = "Pair_ID",
  time = "Year_0",
  fixed_effects = TRUE,
  effect = "section"
)

# Standardization only (no panel identifiers)
X <- data_preparation(df, standardize = TRUE)
```

---

eba *Extreme bounds analysis summaries from a Bayesian model space*

Description

Computes Extreme Bounds Analysis (EBA) summaries for the intercept and each regressor across a model space. For each coefficient, the function reports: the minimum coefficient ("Low"), maximum coefficient ("High"), the mean coefficient ("Mean_coef"), and corresponding "extreme bounds" defined as $\mathrm{Low} - 2 \cdot \mathrm{SE}$ and $\mathrm{High} + 2 \cdot \mathrm{SE}$, where $\mathrm{SE} = \sqrt{\mathrm{VAR}}$ is the standard error associated with the coefficient estimate in the model attaining the minimum/maximum.

Usage

```
eba(betas, VAR, Reg_ID, var_tol = 0)
```

Arguments

- `betas` Numeric matrix of dimension `MS x (K+1)` containing estimated coefficients across models. Column 1 corresponds to the intercept, columns 2 to `K+1` correspond to regressors.
- `VAR` Numeric matrix of dimension `MS x (K+1)` containing variances of the coefficient estimates. Must have the same dimensions as `betas`.
- `Reg_ID` Numeric or integer matrix of dimension `MS x K` indicating regressor inclusion. Entry `Reg_ID[i,k]=1` if regressor `k` is included in model `i`, and 0 otherwise.
- `var_tol` Nonnegative numeric scalar used as a tolerance when checking variance positivity. Entries with `VAR <= var_tol` are treated as invalid for bound calculations. Default is `0`.

Details

The intercept (constant) is assumed to be included in all models. Each regressor is summarized only over models in which it is included, as indicated by the model-inclusion matrix `Reg_ID`.

Value

A numeric matrix of dimension `(K+1) x 5` with columns:

Lower_bound $\min(\beta) - 2 \cdot \mathrm{SE}$ evaluated at the model where β is minimal.

Low Minimum coefficient value across relevant models.

Mean_coef Mean coefficient across relevant models (intercept: all models; regressor: included models only).

High Maximum coefficient value across relevant models.

Upper_bound $\max(\beta) + 2 \cdot \mathrm{SE}$ evaluated at the model where β is maximal.

Rows correspond to the intercept (row 1) and regressors (rows 2..K+1). If a regressor is never included (no 1s in its column of `Reg_ID`), its row will contain `NA`.

`fast_ols` *OLS calculation with additional objects*

Description

OLS calculation with additional objects

Usage

```
fast_ols(y, x)
```

Arguments

- `y` A vector with the dependent variable.
- `x` A matrix with with regressors as columns.

Value

A list with OLS objects: Coefficients, Standard errors, Marginal likelihood, R^2, Degrees of freedom, Determinant of the regressors' matrix.

Examples

```
x1<-rnorm(10, mean = 0, sd = 1)
x2<-rnorm(10, mean = 0, sd = 2)
e<-rnorm(10, mean = 0, sd = 0.5)
y<-2+x1+2*x2+e
x<-cbind(x1,x2)
fast_ols(y,x)
```

`fast_ols_const` *OLS calculation with additional objects for model with just a constant.*

Description

OLS calculation with additional objects for model with just a constant.

Usage

```
fast_ols_const(y)
```

Arguments

y	A vector with the dependent variable.

Value

A list with OLS objects: Coefficients, Standard errors, Marginal likelihood, R^2, Degrees of freedom, Determinant of the regressors matrix.

Examples

```
x1<-rnorm(10, mean = 0, sd = 1)
x2<-rnorm(10, mean = 0, sd = 2)
e<-rnorm(10, mean = 0, sd = 0.5)
y<-2+x1+2*x2+e
x<-cbind(x1,x2)
fast_ols_const(y)
```

`fast_ols_HC` *OLS calculation with heteroscedasticity consistent covariance matrix (MacKinnon & White 1985).*

Description

OLS calculation with heteroscedasticity consistent covariance matrix (MacKinnon & White 1985).

Usage

```
fast_ols_HC(y, x)
```

Arguments

y	A vector with the dependent variable.
x	A matrix with with regressors as columns.

Value

A list with OLS objects: Coefficients, Standard errors, Marginal likelihood, R^2, Degrees of freedom, Determinant of the regressors' matrix.

Examples

```
x1<-rnorm(10, mean = 0, sd = 1)
x2<-rnorm(10, mean = 0, sd = 2)
e<-rnorm(10, mean = 0, sd = 0.5)
y<-2+x1+2*x2+e
x<-cbind(x1,x2)
fast_ols_HC(y,x)
```

`group_dilution` *Compute group-based dilution factors for a model space*

Description

Computes a group-based dilution factor for each model (row) in a model-inclusion matrix. Regressors are assigned to "relatedness groups" via `Nar_vec`. For each model and each group, the dilution exponent increases by 1 for every additional regressor from that group beyond the first. The model's dilution factor is the product of group-specific penalties.

Usage

```
group_dilution(Reg_ID, Nar_vec, p)
```

Arguments

- `Reg_ID` An `MS x K` numeric/integer matrix of model indicators. Each row corresponds to a model; each column corresponds to a regressor. Entries should be 0/1 (0 = excluded, 1 = included).
- `Nar_vec` An integer vector of length `K` giving group assignments for each regressor. Use 0 for regressors not belonging to any group. Positive integers (1,2,...) denote group IDs.
- `p` Either:
  - a single numeric scalar in $[0, 1]$ applied to all groups, or
  - a numeric vector in $[0, 1]$ with one entry per group.

  If `p` is a vector, it is matched to groups as follows:
  - If `p` has names, they must match group IDs (e.g., `c("1"=0.7,"2"=0.5)`),
  - otherwise it is assumed to be in the order of `sort(unique(Nar_vec[Nar_vec>0]))`.

Details

Formally, for model i and group h >= 1, let

$$c_{ih} = \sum_{j=1}^{K} \gamma_{ij} I\{g(j) = h\}$$

and

$$D_i = \sum_{h \geq 1} \max(0, c_{ih} - 1).$$

If `p` is a scalar, the dilution factor is

$$p^{D_i}$$

. If `p` is group-specific with values p_h, then

$$\text{NarDilution}_i = \prod_{h \geq 1} p_h^{\max(0, c_{ih} - 1)}.$$

Value

A numeric vector of length `MS` containing the dilution factor for each model.

Examples

```
# Example model space: MS models, K regressors
Reg_ID <- rbind(
  c(0,0,0,0,0),  # null
  c(1,1,0,0,0),  # two from group 1 -> penalty p_1^(1)
  c(1,1,1,0,0),  # three from group 1 -> penalty p_1^(2)
  c(1,1,0,1,1)   # two from group 1 and two from group 2 -> p_1^1 * p_2^1
)
Nar_vec <- c(1,1,1,2,2)

# Scalar p (same for all groups)
```

```
group_dilution(Reg_ID, Nar_vec, p = 0.7)

# Group-specific p (vector in group order: group 1 then group 2)
group_dilution(Reg_ID, Nar_vec, p = c(0.7, 0.5))

# Group-specific p with explicit mapping via names
group_dilution(Reg_ID, Nar_vec, p = c("1"=0.7, "2"=0.5))
```

---

g_regression *Regression with g prior*

Description

The function implements Bayesian regression with g prior (Zellner, 1986)

Usage

```
g_regression(data, g = "UIP")
```

Arguments

data	A matrix with data. The first column is interpreted as with the dependent variable, while the remaining columns are interpreted as regressors.
g	Value for g in the g prior. Either a number above zero specified by the user or: a) "UIP" for Unit Information Prior (Kass and Wasserman, 1995) b) "RIC" for Risk Inflation Criterion (Foster and George, 1994) c) "Benchmark" for benchmark prior of Fernandez, Ley and Steel (2001) d) "HQ" for prior mimicking Hannan-Quinn information criterion e) "rootUIP" for prior given by the square root of Unit Information Prior

Value

A list with g_regression objects:

1. Expected values of coefficients
2. Posterior standard errors
3. Natural logarithm of marginal likelihood
4. R^2 form ols model
5. Degrees of freedom
6. Determinant of the regressors' matrix

Examples

```
x1 <- rnorm(100, mean = 0, sd = 1)
x2 <- rnorm(100, mean = 0, sd = 2)
e <- rnorm(100, mean = 0, sd = 5)
y <- 2 + x1 + 2*x2 + e
data <- cbind(y,x1,x2)
g_result <- g_regression(data, g = "UIP")
g_result[[1]]
g_result[[2]]

x1 <- rnorm(50, mean = 0, sd = 1)
x2 <- rnorm(50, mean = 0, sd = 2)
e <- rnorm(50, mean = 0, sd = 0.5)
y <- 2 + x1 + 2*x2 + e
data <- cbind(y,x1,x2)
g_result <- g_regression(data, g = "benchmark")
g_result[[1]]
g_result[[2]]
```

---

g_regression_fast *Fast regression with g prior*

Description

The function implements Bayesian regression with g prior (Zellner, 1986)

Usage

```
g_regression_fast(data, g = 0.5)
```

Arguments

data	A matrix with data. The first column is interpreted as with the dependent variable, while the remaining columns are interpreted as regressors.
g	Value for g in the g prior. Default value: g = 0.5.

Value

A list with g_regression objects:

1. Expected values of coefficients
2. Posterior standard errors
3. Natural logarithm of marginal likelihood

4. R^2 form ols model
5. Degrees of freedom
6. Determinant of the regressors' matrix

Examples

```
x1 <- rnorm(100, mean = 0, sd = 1)
x2 <- rnorm(100, mean = 0, sd = 2)
e <- rnorm(100, mean = 0, sd = 5)
y <- 2 + x1 + 2*x2 + e
data <- cbind(y,x1,x2)
g_result <- g_regression_fast(data, g = 0.99)
g_result[[1]]
g_result[[2]]

x1 <- rnorm(50, mean = 0, sd = 1)
x2 <- rnorm(50, mean = 0, sd = 2)
e <- rnorm(50, mean = 0, sd = 0.5)
y <- 2 + x1 + 2*x2 + e
data <- cbind(y,x1,x2)
g_result <- g_regression_fast(data, g = 1.1)
g_result[[1]]
g_result[[2]]
```

---

g_regression_fast_const *Fast regression with g prior for a model with just a constant*

Description

The function implements Bayesian regression with g prior (Zellner, 1986) for a model with just a constant

Usage

```
g_regression_fast_const(y, g = 0.5)
```

Arguments

y	A vector with data - only the dependent variable.
g	Value for g in the g prior. Default value: g = 0.5.

Value

A list with g_regression objects:

1. Expected values of coefficients
2. Posterior standard errors
3. Natural logarithm of marginal likelihood
4. R^2 form ols model
5. Degrees of freedom
6. Determinant of the regressors' matrix

Examples

```
x1 <- rnorm(100, mean = 0, sd = 1)
x2 <- rnorm(100, mean = 0, sd = 2)
e <- rnorm(100, mean = 0, sd = 5)
y <- 2 + x1 + 2*x2 + e
g_result <- g_regression_fast_const(y, g = 0.99)
g_result[[1]]
g_result[[2]]

x1 <- rnorm(50, mean = 0, sd = 1)
x2 <- rnorm(50, mean = 0, sd = 2)
e <- rnorm(50, mean = 0, sd = 0.5)
y <- 2 + x1 + 2*x2 + e
g_result <- g_regression_fast_const(y, g = 1.1)
g_result[[1]]
g_result[[2]]
```

---

g_regression_fast_HC — *Fast regression with g prior and with heteroscedasticity consistent covariance matrix (MacKinnon & White 1985).*

Description

The function implements Bayesian regression with g prior (Zellner, 1986)

Usage

```
g_regression_fast_HC(data, g = 0.5)
```

Arguments

`data` A matrix with data. The first column is interpreted as with the dependent variable, while the remaining columns are interpreted as regressors.

`g` Value for g in the g prior. Default value: g = 0.5.

Value

A list with g_regression objects:

1. Expected values of coefficients
2. Posterior standard errors
3. Natural logarithm of marginal likelihood
4. R^2 form ols model
5. Degrees of freedom
6. Determinant of the regressors' matrix

Examples

```
x1 <- rnorm(100, mean = 0, sd = 1)
x2 <- rnorm(100, mean = 0, sd = 2)
e <- rnorm(100, mean = 0, sd = 5)
y <- 2 + x1 + 2*x2 + e
data <- cbind(y,x1,x2)
g_result <- g_regression_fast_HC(data, g = 0.99)
g_result[[1]]
g_result[[2]]

x1 <- rnorm(50, mean = 0, sd = 1)
x2 <- rnorm(50, mean = 0, sd = 2)
e <- rnorm(50, mean = 0, sd = 0.5)
y <- 2 + x1 + 2*x2 + e
data <- cbind(y,x1,x2)
g_result <- g_regression_fast_HC(data, g = 1.1)
g_result[[1]]
g_result[[2]]
```

`jointness` *Calculation of of the jointness measures*

Description

This function calculates four types of the jointness measures based on the posterior model probabilities calculated using binomial and binomial-beta model prior. The four measures are:

1. HCGHM - for Hofmarcher et al. (2018) measure;
2. LS - for Ley & Steel (2007) measure;
3. DW - for Doppelhofer & Weeks (2009) measure;
4. PPI - for posterior probability of including both variables.
   The measures under binomial model prior will appear in a table above the diagonal, and the measure calculated under binomial-beta model prior below the diagonal.

   REFERENCES
   Doppelhofer G, Weeks M (2009) Jointness of growth determinants. Journal of Applied Econometrics., 24(2), 209-244. doi: 10.1002/jae.1046
   Hofmarcher P, Crespo Cuaresma J, Grün B, Humer S, Moser M (2018) Bivariate jointness measures in Bayesian Model Averaging: Solving the conundrum. Journal of Macroeconomics, 57, 150-165. doi: 10.1016/j.jmacro.2018.05.005
   Ley E, Steel M (2007) Jointness in Bayesian variable selection with applications to growth regression. Journal of Macroeconomics, 29(3), 476-493. doi: 10.1016/j.jmacro.2006.12.002

Usage

```
jointness(bma_list, measure = "HCGHM", rho = 0.5, round = 3)
```

Arguments

`bma_list`	bma object (the result of the bma function)
`measure`	Parameter for choosing the measure of jointness: HCGHM - for Hofmarcher et al. (2018) measure; LS - for Ley & Steel (2007) measure; DW - for Doppelhofer & Weeks (2009) measure; PPI - for posterior probability of including both variables.
`rho`	The parameter "rho" (ρ) to be used in HCGHM jointness measure (default rho = 0.5). Works only if HCGHM measure is chosen (Hofmarcher et al. 2018).
`round`	Parameter indicating the decimal place to which the jointness measures should be rounded (default round = 3).

Value

A table with jointness measures for all the pairs of regressors used in the analysis. The results obtained with the binomial model prior are above the diagonal, while the ones obtained with the binomial-beta prior are below.

Examples

```
data("Trade_data", package = "rmsBMA")
data <- Trade_data[,1:10]
modelSpace <- model_space(data, M = 9, g = "UIP")
bma_list <- bma(modelSpace)
jointness_table <- jointness(bma_list)
```

---

migration_panel *Determinants of International Migration in the European Union*

Description

The dataset contains bilateral migration and its economic determinants for 23 European Union countries over the period 1995–2020. Each observation represents a country pair in a given 5 year period, resulting in 353 unique country pairs and four periods (the first period is utilized for lagged data only).

The countries included are: Austria, Belgium, Czechia, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Netherlands, Poland, Portugal, Slovakia, Slovenia, Spain, Sweden and the United Kingdom

The dataset was used in: Afonso, Aves, Beck (2025), *Drivers of migration flows in the European Union: Earnings or unemployment?*, International Labour Review, 164(2), 1-23. doi: 10.16995/ilr.18845

Usage

```
data(migration_panel)
```

Format

A data frame with 1012 rows and 14 variables:

Year_0 Column indication observation period.

Pair_ID Column indication country pair.

Migration The absolute value of the net migration flows scaled by the sum of the population of a given pair of countries.

MigrationLAG The absolute value of the net migration flows scaled by the sum of the population of a given pair of countries lagged by one period (5 years).

Unempl The absolute value of the difference in unemployment rates, averaged over the 5-year period.

Earn The absolute value of the difference in net earnings expressed in PPP, averaged over the 5-year period.

Tax The absolute value of the difference in mean income tax, averaged over the 5-year period.

Social The absolute value of the difference in mean social benefits per person, averaged over the 5-year period.

LNDGEO A natural logarithm of the distance between the capital of a given pair of countries based on the shortest route.

Temp The absolute value of the difference in mean annual temperature, averaged over the 5-year period.

HC The absolute value of the difference in the human capital index (Barro and Lee 2013), averaged over the 5-year period.

GOV The absolute value of the difference in government spending share of GDP, averaged over the 5-year period.

Gini The absolute value of the difference in the Gini coefficient between a pair of countries, averaged over the 5-year period.

FER The absolute value of the difference in the fertility rate between a pair of countries, averaged over the 5-year period.

Corruption The absolute difference in the value of control of corruption measure from the Worldwide Governance Indicator, averaged over the 5-year period.

Crime The absolute value of the difference in the number of intentional homicides per 1 000 inhabitants, averaged over the 5-year period.

Source

Harvard Dataverse. doi:10.7910/DVN/GTOFJB

`modelSpace` *Model space for the Trade_data_small dataset*

Description

A pre-computed model space object obtained using the `model_space` function on the `Trade_data_small` dataset with a maximum of 7 regressors and the Unit Information Prior.

Format

A list of length 5 with the following components:

x_names Character vector containing the names of the regressors.

ols_results Matrix containing the full model space. Each row corresponds to a model specification and includes:

- binary inclusion indicators for regressors,
- estimated coefficients (including intercept),

- standard errors,
- log-marginal likelihood,
- R^2,
- degrees of freedom,
- dilution prior term.

MS Integer. Total number of models in the model space.

M Integer. Maximum number of regressors allowed in each model.

K Integer. Total number of available regressors.

Details

The object contains all possible linear regression models constructed from the available regressors (up to 7 included simultaneously), together with their estimated coefficients, standard errors, log-marginal likelihood values, R-squared statistics, degrees of freedom, and dilution prior components.

The g-prior specification corresponds to the Unit Information Prior (UIP), i.e. $g = 1/m$, where m denotes the sample size.

Source

Generated using:

```
modelSpace <- model_space(Trade_data_small, M = 7, g = "UIP")
```

See Also

model_space

`model_matrix` *Matrix reflecting model space*

Description

The function creates a matrix with ones indicating inclusion and zeros indicating inclusion of a regressor in a model

Usage

```
model_matrix(K, M)
```

Arguments

K	total number of regressors
M	maximum number of regressor in a model

Value

A matrix with ones indicating inclusion and zeros indicating inclusion of a regressor in a model

`model_pmp` *Graphs of the prior and posterior model probabilities for the best individual models*

Description

This function draws four graphs of prior and posterior model probabilities for the best individual models:
a) The results with binomial model prior (based on PMP - posterior model probability)
b) The results with binomial-beta model prior (based on PMP - posterior model probability)
Models on the graph are ordered according to their posterior model probability.

Arguments

`bma_list`	bma_list object (the result of the bma function)
`top`	The number of the best model to be placed on the graphs

Value

A list with three graphs with prior and posterior model probabilities for individual models:

1. The results with binomial model prior (based on PMP - posterior model probability)
2. The results with binomial-beta model prior (based on PMP - posterior model probability)
3. On graph combining the aforementioned graphs

Examples

```
data("Trade_data", package = "rmsBMA")
data <- Trade_data[,1:10]
modelSpace <- model_space(data, M = 9, g = "UIP")
bma_list <- bma(modelSpace)
model_pmps <- model_pmp(bma_list, top = 100)
model_pmps[[1]]
```

model_sizes — *Graphs of the prior and posterior model probabilities of the model sizes*

Description

This function draws four graphs of prior and posterior model probabilities:
a) The results with binomial model prior (based on PMP - posterior model probability)
b) The results with binomial-beta model prior (based on PMP - posterior model probability)

Arguments

`bma_list`	bma_list object (the result of the bma function)

Value

A list with three graphs with prior and posterior model probabilities for model sizes:

1. The results with binomial model prior (based on PMP - posterior model probability)
2. The results with binomial-beta model prior (based on PMP - posterior model probability)
3. One graph combining all the aforementioned graphs

Examples

```
data("Trade_data", package = "rmsBMA")
data <- Trade_data[,1:10]
modelSpace <- model_space(data, M = 9, g = "UIP")
bma_list <- bma(modelSpace)
sizes <- model_sizes(bma_list)
sizes[[1]]
```

model_space — *Calculation of the model space*

Description

This function calculates all possible models with M regressors that can be constructed out of K regressors.

Usage

```
model_space(data, M = NULL, g = "UIP", HC = FALSE)
```

Arguments

- `data` — Data set to work with. The first column is the data for the dependent variable, and the other columns is the data for the regressors.
- `M` — Maximum number of regressor in the estimated models (default is K - total number of regressors).
- `g` — Value for g in the g prior. Either a number above zero specified by the user or:
  a) "UIP" for Unit Information Prior (Kass and Wasserman, 1995)
  b) "RIC" for Risk Inflation Criterion (Foster and George, 1994)
  c) "Benchmark" for benchmark prior of Fernandez, Ley and Steel (2001)
  d) "HQ" for prior mimicking Hannan-Quinn information criterion
  e) "rootUIP" for prior given by the square root of Unit Information Prior
  f) "None" for the case with no g prior and simple ols regression. In this case the marginal likelihood is calculated according to formula proposed by Leamer (1978).
- `HC` — Logical indicator (default = FALSE) specifying whether a heteroscedasticity-consistent covariance matrix should be used for the estimation of standard errors (MacKinnon & White 1985).

Value

A list with model_space objects:

1. x_names - vector with names of the regressors
2. ols_results - table with the model space - contains ols objects for all the estimated models
3. MS - size of the mode space
4. M - maximum number of regressors in a model
5. K- total number of regressors

Examples

```
x1 <- rnorm(20, mean = 0, sd = 1)
x2 <- rnorm(20, mean = 0, sd = 2)
x3 <- rnorm(20, mean = 0, sd = 3)
x4 <- rnorm(20, mean = 0, sd = 1)
x5 <- rnorm(20, mean = 0, sd = 2)
x6 <- rnorm(20, mean = 0, sd = 4)
e <- rnorm(20, mean = 0, sd = 0.5)
y <- 2 + x1 + 2*x2 + e
data <- cbind(y,x1,x2,x3,x4,x5,x6)
modelSpace <- model_space(data, M = 3)
```

ols — *OLS calculation with additional objects*

Description

OLS calculation with additional objects

Usage

```
ols(y, x, const, Norm = NULL)
```

Arguments

y	A vector with the dependent variable.
x	A matrix with with regressors as columns or 0 for model without any regressors.
const	Binary variable: 1 - include a constant in the estimation, 0 - do not include a constant in the estimation.
Norm	A parameter used to correct likelihood function when it gets to close to zero in the case of high number of observations.

Value

A list with OLS objects: Coefficients, Standard errors, Marginal likelihood, R^2, Degrees of freedom, Determinant of the regressors' matrix, log(Marginal likelihood).

Examples

```
x1<-rnorm(10, mean = 0, sd = 1)
x2<-rnorm(10, mean = 0, sd = 2)
y<-2+x1+2*x2
x<-cbind(x1,x2)
const<-1
ols(y,x,const)

x1<-rnorm(10, mean = 0, sd = 1)
x2<-rnorm(10, mean = 0, sd = 2)
e<-rnorm(10, mean = 0, sd = 0.5)
y<-2+x1+2*x2+e
x<-cbind(x1,x2)
const<-1
ols(y,x,const)
```

`plus_pmp_from_pmp` *Posterior probability of a positive coefficient sign (P(+))*

Description

Computes posterior probabilities of a positive coefficient sign, P(+), for the intercept and each regressor by averaging model-specific probabilities across the model space, weighted by posterior model probabilities.

Usage

```
plus_pmp_from_pmp(pmp_uniform, pmp_random, betas, VAR, DF, Reg_ID)
```

Arguments

`pmp_uniform`	Numeric vector of length `MS` containing posterior model probabilities under a uniform model prior.
`pmp_random`	Numeric vector of length `MS` containing posterior model probabilities under a random model prior.
`betas`	Numeric matrix of dimension `MS x (K+1)` containing estimated coefficients for each model. Column 1 corresponds to the intercept, columns 2 to `K+1` correspond to regressors.
`VAR`	Numeric matrix of dimension `MS x (K+1)` containing variances of the coefficient estimates. Must have the same dimensions as `betas`.
`DF`	Numeric vector of length `MS` containing the degrees of freedom associated with each model.
`Reg_ID`	Numeric or integer matrix of dimension `MS x K` indicating regressor inclusion. Entry `Reg_ID[i, k] = 1` if regressor `k` is included in model `i`, and 0 otherwise.

Details

For a given model i and coefficient j, the contribution is

$$p(M_i \mid y) \cdot F_t\left(\frac{\beta_{ij}}{\sqrt{\mathrm{VAR}_{ij}}}; \mathrm{DF}_i\right),$$

where $F_t(\cdot; \mathrm{DF}_i)$ is the CDF of the Student-t distribution with DF_i degrees of freedom.

The intercept is included in all models, while each regressor contributes only in those models in which it is included, as indicated by the model inclusion matrix `Reg_ID`.

The posterior probability of a positive sign for coefficient j is computed as

$$P(\beta_j > 0 \mid y) = \sum_{i \in \mathcal{M}_j} p(M_i \mid y)\, F_t\left(\frac{\beta_{ij}}{\sqrt{\mathrm{VAR}_{ij}}}; \mathrm{DF}_i\right),$$

where $\mathcal{M}_j$ denotes the set of models that include regressor j. For the intercept, $\mathcal{M}_j$ contains all models.

This definition follows the sign-probability interpretation in *Doppelhofer and Weeks (2009).*

Value

A list with two elements:

Plus_PMP_uniform A `(K+1) x 1` numeric matrix containing posterior probabilities of a positive coefficient sign under the uniform model prior. The first row corresponds to the intercept.

Plus_PMP_random A `(K+1) x 1` numeric matrix containing posterior probabilities of a positive coefficient sign under the random model prior. The first row corresponds to the intercept.

References

Doppelhofer, G. and Weeks, M. (2009). *Jointness of growth determinants*. Journal of Applied Econometrics, 24(2), 209–244.

`posterior_dens` *Graphs of the posterior densities of the coefficients*

Description

This function draws graphs of the posterior densities of all the coefficients of interest.

Arguments

`bma_list`	bma object (the result of the bma function)
`prior`	Parameter indicating which model prior should be used for calculations: 1. "binomial" - using binomial model prior (default option) 2. "beta" - using binomial-beta model prior

Value

A list with the graphs of the posterior densities of coefficients for all the considered regressors.

Examples

```
data <- Trade_data[,1:10]
modelSpace <- model_space(data, M = 9, g = "UIP")
bma_list <- bma(modelSpace)
distPlots <- posterior_dens(bma_list, prior = "binomial")
distPlots[[1]]
distPlots[[2]]
```

posterior_sign_certainty
Posterior sign certainty probability

Description

Computes the posterior probability that a regression coefficient has the same sign as its posterior mean, following the sign-certainty measure used in Sala-i-Martin, Doppelhofer and Miller (2004) and Doppelhofer and Weeks (2009).

Usage

```
posterior_sign_certainty(pmp_uniform, pmp_random, betas, VAR, DF, Reg_ID)
```

Arguments

`pmp_uniform`	Numeric vector of length `MS` containing posterior model probabilities under a uniform model prior.
`pmp_random`	Numeric vector of length `MS` containing posterior model probabilities under a random model prior.
`betas`	Numeric matrix of dimension `MS x (K+1)` containing estimated coefficients for each model. Column 1 corresponds to the intercept.
`VAR`	Numeric matrix of dimension `MS x (K+1)` containing variances of the coefficient estimates. Must have the same dimensions as `betas`.
`DF`	Numeric vector of length `MS` giving the degrees of freedom associated with each model.
`Reg_ID`	Numeric or integer matrix of dimension `MS x K` indicating regressor inclusion. Entry `Reg_ID[i, k] = 1` if regressor `k` is included in model `i`, and 0 otherwise.

Details

For each coefficient j, define

$$S_j = \sum_{i=1}^{MS} p(M_i \mid y)\, F_t\left(\frac{\beta_{ij}}{\sqrt{\mathrm{VAR}_{ij}}}; \mathrm{DF}_i\right),$$

where $F_t(\cdot; \mathrm{DF}_i)$ is the CDF of the Student-t distribution with DF_i degrees of freedom.

The posterior sign certainty probability is then defined as

$$p(\mathrm{sign}_j \mid y) = \begin{cases} S_j, & \text{if } \mathrm{sign}(E[\beta_j \mid y]) > 0, \\ 1 - S_j, & \text{if } \mathrm{sign}(E[\beta_j \mid y]) < 0, \\ 0.5, & \text{if } E[\beta_j \mid y] = 0. \end{cases}$$

The intercept is included in all models. For slope coefficients that are excluded from a given model, the contribution to S_j is set to $F_t(0) = 0.5$, reflecting a symmetric distribution centered at zero.

Value

A list with four elements:

PSC_uniform A `(K+1) x 1` numeric matrix containing posterior sign certainty probabilities under the uniform model prior.

PSC_random A `(K+1) x 1` numeric matrix containing posterior sign certainty probabilities under the random model prior.

PostMean_uniform A `(K+1) x 1` numeric matrix of posterior means under the uniform model prior.

PostMean_random A `(K+1) x 1` numeric matrix of posterior means under the random model prior.

References

Sala-i-Martin, X., Doppelhofer, G., and Miller, R. I. (2004). Determinants of long-term growth: A Bayesian averaging of classical estimates. *American Economic Review*, 94(4), 813–835.

Doppelhofer, G. and Weeks, M. (2009). Jointness of growth determinants. *Journal of Applied Econometrics*, 24(2), 209–244.

`subset_design` *Regressors choice based on the inclusion vector*

Description

The function creates a matrix with regressors that should be included in a specific model based on inclusion vector

Usage

```
subset_design(x, model_row)
```

Arguments

`x`	matrix with regressors
`model_row`	inclusion vector - row of a model space matrix

Value

A matrix with regressors to be used in a specific model

`Trade_data` *Determinants of International Trade in the European Union*

Description

The dataset contains bilateral trade and its economic determinants for 26 European Union countries over the period 1995–2015. Each observation represents a country pair, resulting in 325 unique trading pairs.

The countries included are: Austria, Belgium, Bulgaria, Cyprus, Czechia, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, the Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, and the United Kingdom.

The dataset was used in: Beck (2020), *What drives international trade? Robust analysis for the European Union*, Journal of International Studies, 13(3), 68–84.

Usage

```
data(Trade_data)
```

Format

A data frame with 325 rows and 18 variables:

LNTRADE Natural logarithm of bilateral trade between two countries.

B Border dummy (1 if countries share a common border; 0 otherwise).

LNDGEO Natural logarithm of the shortest distance between capital cities.

L Common language dummy (1 if countries share at least one official language; 0 otherwise).

LNRGDPPROD Natural logarithm of the product of real GDPs of the two countries.

RGDPpcDIFF Absolute difference in real GDP per capita.

GOV Absolute difference in government expenditure shares of GDP.

HUMAN Absolute difference in human capital indicators.

CPW Absolute difference in capital per worker.

INFVAR Absolute difference in the standard deviation of inflation rates.

ARABLE Absolute difference in arable land.

ARABLEpw Absolute difference in arable land per worker.

LAND Absolute difference in total land area.

LANDpc Absolute difference in land area per capita.

EPCpc Absolute difference in electricity consumption per capita.

FDI Absolute difference in foreign direct investment flows.

KSI Krugman specialization index (value added, 35 sectors).

BCIDIFF Absolute difference in the Bayesian Corruption Index.

Source

Harvard Dataverse. doi:10.7910/DVN/JMMOEA

`Trade_data_small` *Determinants of International Trade in the European Union*

Description

The dataset contains bilateral trade and its economic determinants for 26 European Union countries over the period 1995–2015. Each observation represents a country pair, resulting in 325 unique trading pairs.

The countries included are: Austria, Belgium, Bulgaria, Cyprus, Czechia, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, the Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, and the United Kingdom.

The dataset was used in: Beck (2020), *What drives international trade? Robust analysis for the European Union*, Journal of International Studies, 13(3), 68–84.

Usage

```
data(Trade_data_small)
```

Format

A data frame with 325 rows and 11 variables:

LNTRADE Natural logarithm of bilateral trade between two countries.

B Border dummy (1 if countries share a common border; 0 otherwise).

LNDGEO Natural logarithm of the shortest distance between capital cities.

L Common language dummy (1 if countries share at least one official language; 0 otherwise).

LNRGDPPROD Natural logarithm of the product of real GDPs of the two countries.

HUMAN Absolute difference in human capital indicators.

INFVAR Absolute difference in the standard deviation of inflation rates.

ARABLE Absolute difference in arable land.

ARABLEpw Absolute difference in arable land per worker.

LAND Absolute difference in total land area.

LANDpc Absolute difference in land area per capita.

Source

Harvard Dataverse. doi:10.7910/DVN/JMMOEA

Index