Package ‘bpvars’

June 16, 2026

Type Package

Title Forecasting with Bayesian Panel Vector Autoregressions

Description Provides Bayesian estimation and forecasting of dynamic panel data using Bayesian Panel Vector Autoregressions with hierarchical prior distributions following the specification by Sanchez-Martinez & Woźniak (2026) <doi:10.48550/arXiv.2606.14143>. The models include country-specific Vector Autoregressions (VARs) that share a global prior distribution that extend the model by Jarociński (2010) <doi:10.1002/jae.1082>. Under this prior expected value, each country's system follows a global VAR with country-invariant parameters. Further flexibility is provided by the hierarchical prior structure that retains the Minnesota prior interpretation for the global VAR and features estimated prior covariance matrices, shrinkage, and persistence levels. Bayesian forecasting is developed for models including exogenous variables, allowing conditional forecasts given the future trajectories of some variables and restricted forecasts assuring that rates are forecasted to stay positive and less than 100. The package implements the model specification, estimation, and forecasting routines, facilitating coherent workflows and reproducibility. It also includes automated pseudo-out-of-sample forecasting and computation of forecasting performance measures. Beautiful plots, informative summary functions, and extensive documentation complement all this. Extraordinary computational speed is achieved thanks to employing frontier econometric and numerical techniques and algorithms written in 'C++'. The 'bpvars' package is aligned regarding objects, workflows, and code structure with the 'R' packages 'bsvars' by Woźniak (2024) <doi:10.32614/CRAN.package.bsvars>, 'bsvarSIGNs' by Wang & Woźniak (2025) <doi:10.32614/CRAN.package.bsvarSIGNs>, and 'bvars' by Liu, Ramirez Hassan, & Woźniak (2026) <doi:10.32614/CRAN.package.bvars> and they constitute an integrated toolset.

Version 2.0

Date 2026-06-16

Maintainer Tomasz Woźniak <wozniak.tom@pm.me>

License GPL (>= 3)

Depends R (>= 4.1.0), bsvars (>= 3.2)

Imports R6, Rcpp (>= 1.0.12), RcppProgress, RcppTN, generics, TruncatedNormal

LinkingTo bsvars, Rcpp, RcppArmadillo, RcppProgress, RcppTN

Suggests knitr, tinytest

URL https://bsvars.org/bpvars/

BugReports https://github.com/bsvars/bpvars/issues

LazyData true

Encoding UTF-8

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Tomasz Woźniak [aut, cre] (ORCID: <https://orcid.org/0000-0003-2212-2378>), Miguel Sanchez-Martinez [ctb], International Labour Organization [cph] (The International Labour Organization should not be held responsible for any issues arising from the use of the 'bpvars' package or from the results obtained with it.)

Repository CRAN

Date/Publication 2026-06-16 06:50:32 UTC

Contents

`bpvars-package` *Forecasting with Bayesian Panel Vector Autoregressions*

Description

Provides Bayesian estimation and forecasting of dynamic panel data using Bayesian Panel Vector Autoregressions with hierarchical prior distributions following the specification by Sanchez-Martinez & Woźniak (2026) doi:10.48550/arXiv.2606.14143. The models include country-specific Vector Autoregressions (VARs) that share a global prior distribution that extend the model by Jarociński (2010) doi:10.1002/jae.1082. Under this prior expected value, each country's system follows a global VAR with country-invariant parameters. Further flexibility is provided by the hierarchical prior structure that retains the Minnesota prior interpretation for the global VAR and features estimated prior covariance matrices, shrinkage, and persistence levels. Bayesian forecasting is developed for models including exogenous variables, allowing conditional forecasts given the future trajectories of some variables and restricted forecasts assuring that rates are forecasted to stay positive and less than 100. The package implements the model specification, estimation, and forecasting routines, facilitating coherent workflows and reproducibility. It also includes automated pseudo-out-of-sample forecasting and computation of forecasting performance measures. Beautiful plots, informative summary functions, and extensive documentation complement all this. Extraordinary computational speed is achieved thanks to employing frontier econometric and numerical techniques and algorithms written in 'C++'. The 'bpvars' package is aligned regarding objects, workflows, and code structure with the R packages 'bsvars' by Woźniak (2024) doi:10.32614/CRAN.package.bsvars, 'bsvarSIGNs' by Wang & Woźniak (2025) doi:10.32614/CRAN.package.bsvarSIGNs, and 'bvars' by Liu, Ramirez Hassan, & Woźniak (2026) doi:10.32614/CRAN.package.bvars and they constitute an integrated toolset. Copyright: 2025 International Labour Organization. The International Labour Organization should not be held responsible for any issues arising from the use of the 'bpvars' package or from the results obtained with it.

Details

The package provides a set of functions for predictive analysis with the Bayesian Hierarchical Panel Vector Autoregression by Sanchez-Martinez & Woźniak (2026).

The Model. The model specification is initiated using function `specify_bvarPANEL` that creates an object of class `BVARPANEL` containing the prior specification, starting values for estimation, data matrices, and the setup of the Monte Carlo Markov Chain sampling algorithm.

The model features country-specific Vector Autoregressive (VAR) equation for `N` dependent variables with `T_c` observations for each country `c`. Its equation is given by

$$\mathbf{Y}_c = \mathbf{A}_c\mathbf{X}_c + \mathbf{E}_c$$

where $\mathbf{Y}_c$ is an `T_c x N` matrix of dependent variables for country `c`, $\mathbf{X}_c$ is a `T_c x K` matrix of explanatory variables, $\mathbf{E}_c$ is an `T_c x N` matrix of error terms, and $\mathbf{A}_c$ is an `NxK` matrix of country-specific autoregressive slope coefficients and parameters on deterministic terms in $\mathbf{X}_c$. The parameter matrix $\mathbf{A}_c$ includes autoregressive matrices capturing the effects of the lagged vectors of dependent variables at lags from 1 to `p`, a constant term and a set of exogenous variables.

The error terms for each of the periods have zero conditional mean and conditional covariance given by the NxN matrix $\mathbf{\Sigma}$. The errors are jointly normally distributed and serially uncorrelated. These assumptions are summarised using a matrix-variate normal distribution (see Woźniak, 2016):

$$\mathbf{E}_c \sim MN(\mathbf{0}_{T_cxN}, \mathbf{\Sigma}, \mathbf{I}_{T_c})$$

where the identity matrix $\mathbf{I}_{T_c}$ of order T_c and joint normality imply no serial autocorrelation. Matrix $\mathbf{0}_{T_cxN}$ denotes a T_c x N matrix of zeros.

Global Prior Distributions. The Hierarchical Panel VAR model features a sophisticated hierarchical prior structure that grants the model flexibility, interpretability, and improved forecasting performance.

The country-specific parameters follow a prior distribution that, at its mean value, represents a global VAR model with a global autoregressive parameter matrix $\mathbf{A}$ of dimension KxN and an NxN global covariance matrix $\mathbf{\Sigma}$:

$$\mathbf{Y}_c = \mathbf{A}\mathbf{X}_c + \mathbf{E}_c$$

This global VAR model under the prior mean is represented by the parameters of the matrix-variate normal inverted Wishart distribution (see Woźniak, 2016) given by:

$$\mathbf{A}_c, \mathbf{\Sigma}_c | \mathbf{A}, \mathbf{V}, \mathbf{\Sigma}, \nu \sim MNIW(\mathbf{A}, \mathbf{V}, (N - \nu - 1)\mathbf{\Sigma}, \nu)$$

where V is a KxK column-specific covariance matrix, $(N - \nu - 1)\mathbf{\Sigma}$ is the row-specific matrix, and $\nu > N + 1$ is the degrees-of-freedom parameter.

All of the parameters of the prior distribution above feature their own prior distributions and are estimated. These prior distributions are given by:

$$\mathbf{A} | \mathbf{V}, m, w, s \sim MN(m\underline{\mathbf{M}}, \mathbf{V}, s\underline{\mathbf{S}})$$

with the KxN mean matrix $m\underline{\mathbf{M}}$, the KxK column-specific covariance matrix $\mathbf{V}$, and the NxN matrix of row-specific covariance $s\underline{\mathbf{S}}$.

The global error term covariance matrix, $\mathbf{\Sigma}$, follows a Wishart distribution with NxN scale matrix $s\underline{\mathbf{S}}_\Sigma$ and shape parameter $\underline{\mu}_\Sigma$

$$\mathbf{\Sigma} | s, \nu \sim W(s\underline{\mathbf{S}}_\Sigma, \underline{\mu}_\Sigma)$$

Other Prior Distributions. The column-specific covariance $\mathbf{V}$ follows the inverse-Wishart distribution with scale $w\underline{\mathbf{W}}$ and shape $\underline{\eta}$:

$$\mathbf{V} | w \sim IW(w\underline{\mathbf{W}}, \underline{\eta})$$

The shape parameter ν follows an exponential distribution with mean $\underline{\lambda}$:

$$\nu \sim \exp(\underline{\lambda})$$

Finally, the priors for the remaining scalar hyper-parameters are:

$$m \sim N(\underline{\mu}_m, \underline{\sigma}_m^2)$$

$$w \sim G(\underline{s}_w, \underline{a}_w)$$

$$s \sim IG2(\underline{s}_s, \underline{\nu}_s)$$

The prior hyper-parameters in this note are grouped into those that are:

fixed and denoted using underscore, such as e.g. $\underline{\mathbf{M}}$, $\underline{\mu}_{\Sigma}$, or $\underline{\nu}_s$. These hyper-parameters must be fixed and their default values are set by initiating the model specification using function `specify_bvarPANEL`. These values can be accesses from such generated object in its element `prior` and can be modified by the user.

estimated not featuring the underscore in the notation, such as e.g. $\mathbf{A}$, $\mathbf{\Sigma}$, or m. These hyper-parameters are estimated and their posterior draws are available from an object generated after the estimation running the function `estimate`.

Estimation. The package implements Bayesian estimation using the Gibbs sampler. This algorithm provides a sample of random draws from the posterior distribution of the parameters of the model. The posterior distribution is defined by Bayes' Rule stating that the posterior distribution of the parameters given data and is proportional to the likelihood function and the prior distribution of the parameters:

$$p(\theta|\mathbf{Y}) \propto L(\theta; \mathbf{Y})p(\theta)$$

where θ collects all the parameters of the model to be estimated. At each of its iterations a single draw of all of the parameters of the model, including the estimated hyper-parameters, is obtained. This Bayesian procedure estimates jointly all the parameters of the model and is implemented in the `estimate.BVARPANEL` and `estimate.PosteriorBVARPANEL` functions.

Forecasting. The package implements Bayesian forecasting providing the a sample of draws from the joint predictive density defined as the joint density of the future unknown values to be predicted, $\mathbf{Y}_f$, given data, $\mathbf{Y}$ closely following Karlsson (2013):

$$p(\mathbf{Y}_f|\mathbf{Y})$$

The package offers the possibility of:

forecasting for models with exogenous variables given the provided future values of the exogenous variables.

conditional predictions given provided future projections for some of the variables.

trucated forecasts for variables that represents rates from the interval $[0, 100]$.

The forecasting is performed using function `forecast.PosteriorBVARPANEL`.

Note

This package is currently in active development.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Jarociński, M. (2010). Responses to Monetary Policy Shocks in the East and the West of Europe: a Comparison. Journal of Applied Econometrics, 25(5), 833-868, doi:10.1002/jae.1082.

Karlsson, S. (2013). Forecasting with Bayesian Vector Autoregression, in: *Handbook of Economic Forecasting*, Elsevier. volume **2**, 791–897, doi:10.1016/B9780444627315.000154.

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

Woźniak, T. (2016). Bayesian Vector Autoregressions, *Australian Economic Review*, **49**, 365-380, doi:10.1111/14678462.12179.

See Also

`specify_bvarPANEL`, `estimate.BVARPANEL`, `forecast.PosteriorBVARPANEL`

Examples

```
# Basic estimation and forecasting example
############################################################
specification = specify_bvarPANEL$new(ilo_dynamic_panel[1:5]) # specify the model
burn_in      = estimate(specification, S = 5)        # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, S = 5)             # estimate the model; use say S = 10000
predictive    = forecast(posterior, 2)                   # forecast the future

# workflow with the pipe |>
ilo_dynamic_panel[1:5] |>
  specify_bvarPANEL$new() |>
  estimate(S = 5) |>
  estimate(S = 5) |>
  forecast(horizon = 2) -> predictive
```

---

compute_forecast_performance

Computes forecasting performance measures for recursive pseudo-out-of-sample forecasts

---

Description

Computes forecasting performance measures selected from: log-predictive score `"lps"`, root-mean-squared-forecast error `"rmsfe"`, mean-absolute-forecast error `"mafe"` from the output of the recursive pseudo-out-of-sample forecastinge exercise performed using function `forecast_poos_recursively`. See Sanchez-Martinez & Woźniak (2026).

Usage

```
compute_forecast_performance(forecasts, measures = c("pls", "rmsfe", "mafe"))
```

Arguments

`forecasts`	an object containing the outcome of Bayesian recursive pseudo-out-of-sample forecasting exercise using expanding window samples generated using function `forecast_poos_recursively`.
`measures`	a character vector with any of the values `"lps"`, `"rmsfe"`, `"mafe"` indicating the forecasting performance measures to be computed.

Value

An object of class `ForecastingPerformance`

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`forecast_poos_recursively`, `forecast_poos_recursively.BVARPANEL`, `forecast_poos_recursively.BVARGROUPPAN`

Examples

```
spec = specify_bvarPANEL$new(ilo_dynamic_panel[1:5])              # specify the model
poos = specify_poosf_exercise$new(spec, 2, 5, 1, 30)   # specify the forecasting  exercise
fore = forecast_poos_recursively(spec, poos)           # perform the forecasting  exercise
fp  = compute_forecast_performance(fore, "pls")  # compute forecasting performance measures
```

`compute_forecast_performance.ForecastsPANELpoos`

Computes forecasting performance measures for recursive pseudo-out-of-sample forecasts

---

Description

Computes forecasting performance measures selected from: log-predictive score `"lps"`, root-mean-squared-forecast error `"rmsfe"`, mean-absolute-forecast error `"mafe"` from the output of the recursive pseudo-out-of-sample forecastinge exercise performed using function `forecast_poos_recursively`. See Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'ForecastsPANELpoos'
compute_forecast_performance(forecasts, measures = c("pls", "rmsfe", "mafe"))
```

Arguments

`forecasts`	an object of class `ForecastsPANELpoos` containing the outcome of Bayesian recursive pseudo-out-of-sample forecasting exercise using expanding window samples generated using function `forecast_poos_recursively`.
`measures`	a character vector with any of the values `"lps"`, `"rmsfe"`, `"mafe"` indicating the forecasting performance measures to be computed.

Value

An object of class `ForecastingPerformance`

Author(s)

Tomasz Woźniak <`wozniak.tom@pm.me`>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`forecast_poos_recursively`, `forecast_poos_recursively.BVARPANEL`, `forecast_poos_recursively.BVARGROUPPAN`

Examples

```
spec = specify_bvarPANEL$new(ilo_dynamic_panel[1:5])              # specify the model
poos = specify_poosf_exercise$new(spec, 2, 5, 1, 30)   # specify the forecasting  exercise
fore = forecast_poos_recursively(spec, poos)          # perform the forecasting  exercise
fp  = compute_forecast_performance(fore, "pls")  # compute forecasting performance measures
```

`compute_variance_decompositions.PosteriorBVARGROUPPANEL`
Computes posterior draws of the forecast error variance decomposition

---

Description

For each country, each of the draws from the posterior estimation of the model is transformed into a draw from the posterior distribution of the forecast error variance decomposition. See Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'PosteriorBVARGROUPPANEL'
compute_variance_decompositions(posterior, horizon)
```

Arguments

`posterior`	posterior estimation outcome - an object of class `PosteriorBVARGROUPPANEL` obtained by running the `estimate` function.
`horizon`	a positive integer number denoting the forecast horizon for the forecast error variance decompositions.

Value

An object of class `PosteriorFEVDPANEL`, that is, a list with `C` elements containing `NxNx(horizon+1)xS` arrays of class `PosteriorFEVD` with `S` draws of country-specific forecast error variance decompositions.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Lütkepohl, H. (2017). Structural VAR Tools, Chapter 4, In: Structural vector autoregressive analysis. Cambridge University Press.

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`estimate.PosteriorBVARGROUPPANEL`, `summary.PosteriorFEVDPANEL`, `plot.PosteriorFEVDPANEL`

Examples

```
# specify the model and set seed
specification  = specify_bvarGroupPANEL$new(            # specify the model
                   ilo_dynamic_panel[1:5],
                   group_allocation = country_grouping_region[1:5]
                 )

# run the burn-in
burn_in        = estimate(specification, 5)

# estimate the model
posterior      = estimate(burn_in, 5)

# compute forecast error variance decomposition 4 years ahead
fevd           = compute_variance_decompositions(posterior, horizon = 4)
```

`compute_variance_decompositions.PosteriorBVARPANEL`

Computes posterior draws of the forecast error variance decomposition

---

Description

For each country, each of the draws from the posterior estimation of the model is transformed into a draw from the posterior distribution of the forecast error variance decomposition. See Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'PosteriorBVARPANEL'
compute_variance_decompositions(posterior, horizon)
```

Arguments

`posterior`	posterior estimation outcome - an object of class `PosteriorBVARPANEL` obtained by running the `estimate` function.
`horizon`	a positive integer number denoting the forecast horizon for the forecast error variance decompositions.

Value

An object of class `PosteriorFEVDPANEL`, that is, a list with `C` elements containing `NxNx(horizon+1)xS` arrays of class `PosteriorFEVD` with `S` draws of country-specific forecast error variance decompositions.

Author(s)

Tomasz Woźniak `<wozniak.tom@pm.me>`

References

Lütkepohl, H. (2017). Structural VAR Tools, Chapter 4, In: Structural vector autoregressive analysis. Cambridge University Press.

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`estimate.PosteriorBVARPANEL`, `summary.PosteriorFEVDPANEL`, `plot.PosteriorFEVDPANEL`

Examples

```
# specify the model and set seed
specification  = specify_bvarPANEL$new(ilo_dynamic_panel[1:5], p = 1)

# run the burn-in
burn_in        = estimate(specification, 5)

# estimate the model
posterior      = estimate(burn_in, 5)

# compute forecast error variance decomposition 4 years ahead
fevd           = compute_variance_decompositions(posterior, horizon = 4)
```

compute_variance_decompositions.PosteriorBVARs
Computes posterior draws of the forecast error variance decomposition

Description

For each country, each of the draws from the posterior estimation of the model is transformed into a draw from the posterior distribution of the forecast error variance decomposition. See Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'PosteriorBVARs'
compute_variance_decompositions(posterior, horizon)
```

Arguments

`posterior`	posterior estimation outcome - an object of class `PosteriorBVARs` obtained by running the `estimate` function.
`horizon`	a positive integer number denoting the forecast horizon for the forecast error variance decompositions.

Value

An object of class `PosteriorFEVDPANEL`, that is, a list with `C` elements containing `NxNx(horizon+1)xS` arrays of class `PosteriorFEVD` with `S` draws of country-specific forecast error variance decompositions.

Author(s)

Tomasz Woźniak `<wozniak.tom@pm.me>`

References

Lütkepohl, H. (2017). Structural VAR Tools, Chapter 4, In: Structural vector autoregressive analysis. Cambridge University Press.

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`estimate.PosteriorBVARs`, `summary.PosteriorFEVDPANEL`, `plot.PosteriorFEVDPANEL`

Examples

```
# specify the model and set seed
specification  = specify_bvars$new(ilo_dynamic_panel[1:5]) # specify the model

# run the burn-in
burn_in        = estimate(specification, 5)

# estimate the model
posterior      = estimate(burn_in, 5)

# compute forecast error variance decomposition 4 years ahead
fevd           = compute_variance_decompositions(posterior, horizon = 4)
```

---

country_grouping_incomegroup

A vector with country grouping by income group for 189 countries

---

Description

Each of the country is classified into one of the 4 categories according to their geographical location. The categories are:

1 Low-income countries

2 Lower-middle-income countries

3 Upper-middle-income countries

4 High-income countries

Last data update was implemented on 2026-04-18.

Usage

```
data(country_grouping_incomegroup)
```

Format

A numeric vector with values from 1 to 4

Source

International Labour Organization. (2020). ILO modelled estimates database, ILOSTAT [database]. Available from https://ilostat.ilo.org/data/.

Examples

```
data(country_grouping_incomegroup)   # upload the data

# setup a fixed group allocation Panel VAR model
spec = specify_bvarGroupPANEL$new(
         ilo_dynamic_panel,
         group_allocation = country_grouping_incomegroup
)
```

---

country_grouping_region

A vector with country grouping by region for 189 countries

Description

Each of the country is classified into one of the 5 categories according to their geographical location. The categories are:

1 Asia and the Pacific

2 Africa

3 Europe and Central Asia

4 Arab States

5 Americas

Last data update was implemented on 2026-04-18.

Usage

```
data(country_grouping_region)
```

Format

A numeric vector with values from `1` to `5`

Source

International Labour Organization. (2020). ILO modelled estimates database, ILOSTAT [database]. Available from https://ilostat.ilo.org/data/.

Examples

```
data(country_grouping_region)   # upload the data

# setup a fixed group allocation Panel VAR model
spec = specify_bvarGroupPANEL$new(
         ilo_dynamic_panel,
         group_allocation = country_grouping_region
)
```

`country_grouping_subregionbroad`
A vector with country grouping by subregion for 189 countries

Description

Each of the country is classified into one of the 11 categories according to their geographical location. The categories are:

1 Southern Asia

2 Sub-Saharan Africa

3 Northern, Southern and Western Europe

4 Arab States

5 Latin America and the Caribbean

6 Central and Western Asia

7 South-Eastern Asia and the Pacific

8 Eastern Europe

9 Northern America

10 Eastern Asia

11 Northern Africa

Last data update was implemented on 2026-04-18.

Usage

```
data(country_grouping_subregionbroad)
```

Format

A numeric vector with values from 1 to 11

Source

International Labour Organization. (2020). ILO modelled estimates database, ILOSTAT [database]. Available from https://ilostat.ilo.org/data/.

Examples

```
data(country_grouping_subregionbroad)   # upload the data

# setup a fixed group allocation Panel VAR model
spec = specify_bvarGroupPANEL$new(
         ilo_dynamic_panel,
         group_allocation = country_grouping_subregionbroad
)
```

`country_grouping_subregiondetailed`
A vector with country grouping by detailed subregion for 189 countries

---

Description

Each of the country is classified into one of the 20 categories according to their geographical location. The categories are:

1 Southern Asia

2 Central Africa

3 Southern Europe

4 Arab States

5 South America

6 Western Asia

7 Pacific Islands

8 Western Europe

9 Eastern Africa

10 Western Africa

11 Eastern Europe

12 Caribbean

13 Central America

14 South-Eastern Asia

15 Southern Africa

16 Northern America

17 Northern Europe

18 Eastern Asia

19 Northern Africa

20 Central Asia

Last data update was implemented on 2026-04-18.

Usage

```
data(country_grouping_subregiondetailed)
```

Format

A numeric vector with values from `1` to `20`

Source

International Labour Organization. (2020). ILO modelled estimates database, ILOSTAT [database]. Available from https://ilostat.ilo.org/data/.

Examples

```
data(country_grouping_subregiondetailed)   # upload the data

# setup a fixed group allocation Panel VAR model
spec = specify_bvarGroupPANEL$new(
         ilo_dynamic_panel,
         group_allocation = country_grouping_subregiondetailed
)
```

`estimate.BVARGROUPPANEL`

Bayesian estimation of a Bayesian Hierarchical Panel Vector Autoregression with fixed or estimated country grouping

---

Description

Estimates the Bayesian Hierarchical Panel VAR with fixed or estimated country grouping using the Gibbs sampler proposed by Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'BVARGROUPPANEL'
estimate(specification, S, thin = 1L, show_progress = TRUE)
```

Arguments

`specification`	an object of class `BVARGROUPPANEL` generated using the `specify_bvarPANEL$new()` function.
`S`	a positive integer, the number of posterior draws to be generated
`thin`	a positive integer, specifying the frequency of MCMC output thinning
`show_progress`	a logical value, if `TRUE` the estimation progress bar is visible

Details

The Bayesian Hierarchical Panel Vector Autoregressive model with fixed or estimated country grouping is estimated using the Gibbs sampler. In this estimation procedure all the parameters of the model are estimated jointly. The list of parameters of the model includes:

$\mathbf{A}_c$ a `KxN` country-specific autoregressive parameters matrix for each of the countries $c = 1, \dots, C$

$\mathbf{\Sigma}_c$ an `NxN` country-specific covariance matrix for each of the countries $c = 1, \dots, C$

$\mathbf{A}$ a `KxN` global autoregressive parameters matrix

$\mathbf{\Sigma}$ an NxN global covariance matrix

$\mathbf{V}$ a KxK covariance matrix of prior for global autoregressive parameters

ν prior degrees of freedom parameter

m prior average global persistence parameter

w prior scaling parameter

s prior scaling parameter

Parameters $\mathbf{A}_c$ and $\mathbf{\Sigma}_c$ are subject to country grouping which sets them to the group-specific values.

Gibbs sampler is an algorithm to sample random draws from the posterior distribution of the parameters of the model given the data. The algorithm is briefly explained on an example of a two-parameter model with parameters θ_1 and θ_2. In order to sample from the joint posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$ the Gibbs sampler proceeds by sampling from full-conditional posterior distributions of each parameter given data and all the other parameters, denoted by $p(\theta_1|\theta_2, \mathbf{Y})$ and $p(\theta_2|\theta_1, \mathbf{Y})$. These distributions are available from derivations and should be in a form of distributions that are easy to sample random numbers from.

To obtain `S` draws from the posterior distribution:

1. Set the initial values of the parameters $\theta_2^{(0)}$
2. At each of the `s` iterations:
   (a) Sample $\theta_1^{(s)}$ from $p(\theta_1|\theta_2^{(s-1)}, \mathbf{Y})$
   (b) Sample $\theta_2^{(s)}$ from $p(\theta_2|\theta_1^{(s)}, \mathbf{Y})$
3. Repeat step 2. `S` times. Return $\{\theta_1^{(s)}, \theta_2^{(s)}\}_{s=1}^{S}$ as a sample drawn from the posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$.

The `estimate()` function returns the draws from the posterior distribution of the parameters of the hierarchical panel VAR model listed above.

Thinning. Thinning is a procedure to reduce the dependence in the returned sample from the posterior distribution. It is obtained by returning every `thin` draw in the final sample. This procedure reduces the number of draws returned by the `estimate()` function.

Value

An object of class `PosteriorBVARGROUPPANEL` containing the Bayesian estimation output and containing two elements:

`posterior` a list with a collection of `S` draws from the posterior distribution generated via Gibbs sampler. Elements of the list correspond to the parameters of the model listed in section **Details** and are named respectively: `A_c`, `Sigma_c`, `A`, `Sigma`, `V`, `nu`, `m`, `w`, `s`.

`last_draw` an object of class `BVARGROUPPANEL` with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using the `estimate()` method.

Author(s)

Tomasz Woźniak `<wozniak.tom@pm.me>`

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`bpvars`, `specify_bvarGroupPANEL`, `specify_posterior_bvarGroupPANEL`, `summary.PosteriorBVARGROUPPANEL`, `forecast.PosteriorBVARGROUPPANEL`

Examples

```
# specify the model
specification = specify_bvarGroupPANEL$new(
     data = ilo_dynamic_panel[1:5],
     exogenous = ilo_exogenous_variables[1:5],
     group_allocation = country_grouping_region[1:5]
)
burn_in      = estimate(specification, 10)          # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 10)             # estimate the model; use say S = 10000
```

`estimate.BVARGROUPPRIORPANEL`

Bayesian estimation of a Bayesian Hierarchical Panel Vector Autoregression with fixed or estimated country grouping for global priors

---

Description

Estimates the Bayesian Hierarchical Panel VAR with fixed or estimated country grouping for global prior parameters using the Gibbs sampler proposed by Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'BVARGROUPPRIORPANEL'
estimate(specification, S, thin = 1L, show_progress = TRUE)
```

Arguments

`specification`	an object of class `BVARGROUPPRIORPANEL` generated using the `specify_bvarGroupPriorPANEL$new()` function.
`S`	a positive integer, the number of posterior draws to be generated
`thin`	a positive integer, specifying the frequency of MCMC output thinning
`show_progress`	a logical value, if `TRUE` the estimation progress bar is visible

Details

The Bayesian Hierarchical Panel Vector Autoregressive model with fixed or estimated country grouping for global prior parametrs is estimated using the Gibbs sampler. In this estimation procedure all the parameters of the model are estimated jointly. The list of parameters of the model includes:

$\mathbf{A}_c$ a `KxN` country-specific autoregressive parameters matrix for each of the countries $c = 1, \dots, C$

$\mathbf{\Sigma}_c$ an `NxN` country-specific covariance matrix for each of the countries $c = 1, \dots, C$

$\mathbf{A}_g$ a `KxNxG` group-specific global autoregressive parameters matrix

$\mathbf{\Sigma}_g$ an `NxN` group-specific global covariance matrix

$\mathbf{V}$ a `KxK` covariance matrix of prior for global autoregressive parameters

ν prior degrees of freedom parameter

m prior average global persistence parameter

w prior scaling parameter

s prior scaling parameter

Parameters $\mathbf{A}_c$ and $\mathbf{\Sigma}_c$ have prior expected values determined by the group-specific prior parameters $\mathbf{A}_g$ and $\mathbf{\Sigma}_g$.

Gibbs sampler is an algorithm to sample random draws from the posterior distribution of the parameters of the model given the data. The algorithm is briefly explained on an example of a two-parameter model with parameters θ_1 and θ_2. In order to sample from the joint posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$ the Gibbs sampler proceeds by sampling from full-conditional posterior distributions of each parameter given data and all the other parameters, denoted by $p(\theta_1|\theta_2, \mathbf{Y})$ and $p(\theta_2|\theta_1, \mathbf{Y})$. These distributions are available from derivations and should be in a form of distributions that are easy to sample random numbers from.

To obtain `S` draws from the posterior distribution:

1. Set the initial values of the parameters $\theta_2^{(0)}$
2. At each of the `s` iterations:
   (a) Sample $\theta_1^{(s)}$ from $p(\theta_1|\theta_2^{(s-1)}, \mathbf{Y})$
   (b) Sample $\theta_2^{(s)}$ from $p(\theta_2|\theta_1^{(s)}, \mathbf{Y})$
3. Repeat step 2. `S` times. Return $\{\theta_1^{(s)}, \theta_2^{(s)}\}_{s=1}^{S}$ as a sample drawn from the posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$.

The `estimate()` function returns the draws from the posterior distribution of the parameters of the hierarchical panel VAR model listed above.

Thinning. Thinning is a procedure to reduce the dependence in the returned sample from the posterior distribution. It is obtained by returning every `thin` draw in the final sample. This procedure reduces the number of draws returned by the `estimate()` function.

Value

An object of class `PosteriorBVARGROUPPRIORPANEL` containing the Bayesian estimation output and containing two elements:

`posterior` a list with a collection of `S` draws from the posterior distribution generated via Gibbs sampler. Elements of the list correspond to the parameters of the model listed in section **Details** and are named respectively: `A_c`, `Sigma_c`, `A_g`, `Sigma_g`, `V`, `nu`, `m`, `w`, `s`.

`last_draw` an object of class `BVARGROUPPRIORPANEL` with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using the `estimate()` method.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`bpvars`, `specify_bvarGroupPriorPANEL`, `specify_posterior_bvarGroupPriorPANEL`, `summary.PosteriorBVARGROUPP`
`forecast.PosteriorBVARGROUPPRIORPANEL`

Examples

```
# specify the model
specification = specify_bvarGroupPriorPANEL$new(
     data = ilo_dynamic_panel[1:5],
     exogenous = ilo_exogenous_variables[1:5],
     group_allocation = country_grouping_region[1:5]
)
burn_in      = estimate(specification, 10)          # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 10)              # estimate the model; use say S = 10000
```

`estimate.BVARPANEL` *Bayesian estimation of a Bayesian Hierarchical Panel Vector Autoregression using Gibbs sampler*

Description

Estimates the Bayesian Hierarchical Panel VAR using the Gibbs sampler proposed by Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'BVARPANEL'
estimate(specification, S, thin = 1L, show_progress = TRUE)
```

Arguments

`specification`	an object of class `BVARPANEL` generated using the `specify_bvarPANEL$new()` function.
`S`	a positive integer, the number of posterior draws to be generated
`thin`	a positive integer, specifying the frequency of MCMC output thinning
`show_progress`	a logical value, if `TRUE` the estimation progress bar is visible

Details

The Bayesian Hierarchical Panel Vector Autoregressive model described in `bpvars` is estimated using the Gibbs sampler. In this estimation procedure all the parameters of the model are estimated jointly. The list of parameters of the model includes:

$\mathbf{A}_c$ a `KxN` country-specific autoregressive parameters matrix for each of the countries $c = 1, \dots, C$

$\mathbf{\Sigma}_c$ an `NxN` country-specific covariance matrix for each of the countries $c = 1, \dots, C$

$\mathbf{A}$ a `KxN` global autoregressive parameters matrix

$\mathbf{\Sigma}$ an `NxN` global covariance matrix

$\mathbf{V}$ a `KxK` covariance matrix of prior for global autoregressive parameters

ν prior degrees of freedom parameter

m prior average global persistence parameter

w prior scaling parameter

s prior scaling parameter

Gibbs sampler is an algorithm to sample random draws from the posterior distribution of the parameters of the model given the data. The algorithm is briefly explained on an example of a two-parameter model with parameters θ_1 and θ_2. In order to sample from the joint posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$ the Gibbs sampler proceeds by sampling from full-conditional posterior distributions of each parameter given data and all the other parameters, denoted by $p(\theta_1|\theta_2, \mathbf{Y})$ and $p(\theta_2|\theta_1, \mathbf{Y})$. These distributions are available from derivations and should be in a form of distributions that are easy to sample random numbers from.

To obtain `S` draws from the posterior distribution:

1. Set the initial values of the parameters $\theta_2^{(0)}$
2. At each of the `s` iterations:
   (a) Sample $\theta_1^{(s)}$ from $p(\theta_1|\theta_2^{(s-1)}, \mathbf{Y})$
   (b) Sample $\theta_2^{(s)}$ from $p(\theta_2|\theta_1^{(s)}, \mathbf{Y})$
3. Repeat step 2. `S` times. Return $\{\theta_1^{(s)}, \theta_2^{(s)}\}_{s=1}^{S}$ as a sample drawn from the posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$.

The `estimate()` function returns the draws from the posterior distribution of the parameters of the hierarchical panel VAR model listed above.

Thinning. Thinning is a procedure to reduce the dependence in the returned sample from the posterior distribution. It is obtained by returning every `thin` draw in the final sample. This procedure reduces the number of draws returned by the `estimate()` function.

Value

An object of class `PosteriorBVARPANEL` containing the Bayesian estimation output and containing two elements:

`posterior` a list with a collection of `S` draws from the posterior distribution generated via Gibbs sampler. Elements of the list correspond to the parameters of the model listed in section **Details** and are named respectively: `A_c`, `Sigma_c`, `A`, `Sigma`, `V`, `nu`, `m`, `w`, `s`.

`last_draw` an object of class `BVARPANEL` with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using the `estimate()` method.

Author(s)

Tomasz Woźniak <`wozniak.tom@pm.me`>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`bpvars`, `specify_bvarPANEL`, `specify_posterior_bvarPANEL`, `summary.PosteriorBVARPANEL`, `forecast.PosteriorBVARPANEL`

Examples

```
# specify the model
specification = specify_bvarPANEL$new(
  ilo_dynamic_panel[1:5],
  exogenous = ilo_exogenous_variables[1:5]
)
burn_in      = estimate(specification, 10)           # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 10)              # estimate the model; use say S = 10000
```

`estimate.BVARs`	*Bayesian estimation of a Bayesian Hierarchical Vector Autoregressions for cubic data using Gibbs sampler*

Description

Estimates the Bayesian Hierarchical VARs for cubic data using the Gibbs sampler proposed by Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'BVARs'
estimate(specification, S, thin = 1L, show_progress = TRUE)
```

Arguments

`specification`	an object of class `BVARs` generated using the `specify_bvars$new()` function.
`S`	a positive integer, the number of posterior draws to be generated
`thin`	a positive integer, specifying the frequency of MCMC output thinning
`show_progress`	a logical value, if `TRUE` the estimation progress bar is visible

Details

The Bayesian Hierarchical Vector Autoregressive models described in `bpvars` is estimated using the Gibbs sampler. In this estimation procedure all the parameters of the model are estimated jointly. The list of parameters of the model includes:

$\mathbf{A}_c$ a `KxN` country-specific autoregressive parameters matrix for each of the countries $c = 1, \dots, C$

$\mathbf{\Sigma}_c$ an `NxN` country-specific covariance matrix for each of the countries $c = 1, \dots, C$

ν prior degrees of freedom parameter

m prior average global persistence parameter

w prior scaling parameter

s prior scaling parameter

Gibbs sampler is an algorithm to sample random draws from the posterior distribution of the parameters of the model given the data. The algorithm is briefly explained on an example of a two-parameter model with parameters θ_1 and θ_2. In order to sample from the joint posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$ the Gibbs sampler proceeds by sampling from full-conditional posterior distributions of each parameter given data and all the other parameters, denoted by $p(\theta_1|\theta_2, \mathbf{Y})$ and $p(\theta_2|\theta_1, \mathbf{Y})$. These distributions are available from derivations and should be in a form of distributions that are easy to sample random numbers from.

To obtain `S` draws from the posterior distribution:

1. Set the initial values of the parameters $\theta_2^{(0)}$
2. At each of the `s` iterations:

   (a) Sample $\theta_1^{(s)}$ from $p(\theta_1|\theta_2^{(s-1)}, \mathbf{Y})$
   (b) Sample $\theta_2^{(s)}$ from $p(\theta_2|\theta_1^{(s)}, \mathbf{Y})$
3. Repeat step 2. `S` times. Return $\{\theta_1^{(s)}, \theta_2^{(s)}\}_{s=1}^{S}$ as a sample drawn from the posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$.

The `estimate()` function returns the draws from the posterior distribution of the parameters of the hierarchical VAR model listed above.

Thinning. Thinning is a procedure to reduce the dependence in the returned sample from the posterior distribution. It is obtained by returning every `thin` draw in the final sample. This procedure reduces the number of draws returned by the `estimate()` function.

Value

An object of class `PosteriorBVARs` containing the Bayesian estimation output and containing two elements:

`posterior` a list with a collection of `S` draws from the posterior distribution generated via Gibbs sampler. Elements of the list correspond to the parameters of the model listed in section **Details** and are named respectively: `A_c`, `Sigma_c`, `nu`, `m`, `w`, `s`.

`last_draw` an object of class `BVARs` with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using the `estimate()` method.

Author(s)

Tomasz Woźniak <`wozniak.tom@pm.me`>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`bpvars`, `specify_bvars`, `specify_posterior_bvars`, `summary.PosteriorBVARs`, `forecast.PosteriorBVARs`

Examples

```
# specify the model
specification = specify_bvars$new(
  ilo_dynamic_panel[1:5],
  exogenous = ilo_exogenous_variables[1:5]
)
burn_in      = estimate(specification, 10)          # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 10)              # estimate the model; use say S = 10000
```

`estimate.PosteriorBVARGROUPPANEL`

Bayesian estimation of a Bayesian Hierarchical Panel Vector Autoregression with fixed or estimated country grouping

Description

Estimates the Bayesian Hierarchical Panel VAR with fixed or estimated country grouping using the Gibbs sampler proposed by Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'PosteriorBVARGROUPPANEL'
estimate(specification, S, thin = 1, show_progress = TRUE)
```

Arguments

`specification` an object of class `PosteriorBVARGROUPPANEL` generated using the `estimate.BVARGROUPPANEL()` function. This setup facilitates the continuation of the MCMC sampling starting from the last draw of the previous run.

`S` a positive integer, the number of posterior draws to be generated

`thin` a positive integer, specifying the frequency of MCMC output thinning

`show_progress` a logical value, if `TRUE` the estimation progress bar is visible

Details

The Bayesian Hierarchical Panel Vector Autoregressive model with fixed or estimated country grouping is estimated using the Gibbs sampler. In this estimation procedure all the parameters of the model are estimated jointly. The list of parameters of the model includes:

$\mathbf{A}_c$ a `KxN` country-specific autoregressive parameters matrix for each of the countries $c = 1, \dots, C$

$\mathbf{\Sigma}_c$ an `NxN` country-specific covariance matrix for each of the countries $c = 1, \dots, C$

$\mathbf{A}$ a `KxN` global autoregressive parameters matrix

$\mathbf{\Sigma}$ an `NxN` global covariance matrix

$\mathbf{V}$ a `KxK` covariance matrix of prior for global autoregressive parameters

ν prior degrees of freedom parameter

m prior average global persistence parameter

w prior scaling parameter

s prior scaling parameter

Parameters $\mathbf{A}_c$ and $\mathbf{\Sigma}_c$ are subject to country grouping which sets them to the group-specific values.

Gibbs sampler is an algorithm to sample random draws from the posterior distribution of the parameters of the model given the data. The algorithm is briefly explained on an example of a two-parameter model with parameters θ_1 and θ_2. In order to sample from the joint posterior distribution

$p(\theta_1, \theta_2|\mathbf{Y})$ the Gibbs sampler proceeds by sampling from full-conditional posterior distributions of each parameter given data and all the other parameters, denoted by $p(\theta_1|\theta_2, \mathbf{Y})$ and $p(\theta_2|\theta_1, \mathbf{Y})$. These distributions are available from derivations and should be in a form of distributions that are easy to sample random numbers from.

To obtain `S` draws from the posterior distribution:

1. Set the initial values of the parameters $\theta_2^{(0)}$
2. At each of the `s` iterations:
   (a) Sample $\theta_1^{(s)}$ from $p(\theta_1|\theta_2^{(s-1)}, \mathbf{Y})$
   (b) Sample $\theta_2^{(s)}$ from $p(\theta_2|\theta_1^{(s)}, \mathbf{Y})$
3. Repeat step 2. `S` times. Return $\{\theta_1^{(s)}, \theta_2^{(s)}\}_{s=1}^S$ as a sample drawn from the posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$.

The `estimate()` function returns the draws from the posterior distribution of the parameters of the hierarchical panel VAR model listed above.

Thinning. Thinning is a procedure to reduce the dependence in the returned sample from the posterior distribution. It is obtained by returning every `thin` draw in the final sample. This procedure reduces the number of draws returned by the `estimate()` function.

Value

An object of class `PosteriorBVARGROUPPANEL` containing the Bayesian estimation output and containing two elements:

`posterior` a list with a collection of `S` draws from the posterior distribution generated via Gibbs sampler. Elements of the list correspond to the parameters of the model listed in section **Details** and are named respectively: `A_c`, `Sigma_c`, `A`, `Sigma`, `V`, `nu`, `m`, `w`, `s`.

`last_draw` an object of class `BVARGROUPPANEL` with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using the `estimate()` method.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`bpvars`, `specify_bvarGroupPANEL`, `specify_posterior_bvarGroupPANEL`, `summary.PosteriorBVARGROUPPANEL`, `forecast.PosteriorBVARGROUPPANEL`

Examples

```
# specify the model
specification = specify_bvarGroupPANEL$new(
     data = ilo_dynamic_panel[1:5],
     exogenous = ilo_exogenous_variables[1:5],
     group_allocation = country_grouping_region[1:5]
)
burn_in      = estimate(specification, 10)           # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 10)              # estimate the model; use say S = 10000
```

`estimate.PosteriorBVARGROUPPRIORPANEL`

Bayesian estimation of a Bayesian Hierarchical Panel Vector Autoregression with fixed or estimated country grouping for global priors

---

Description

Estimates the Bayesian Hierarchical Panel VAR with fixed or estimated country grouping for global prior parameters using the Gibbs sampler proposed by Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'PosteriorBVARGROUPPRIORPANEL'
estimate(specification, S, thin = 1, show_progress = TRUE)
```

Arguments

`specification`	an object of class `PosteriorBVARGROUPPRIORPANEL` generated using the `estimate.BVARGROUPPRIORPANEL()` function. This setup facilitates the continuation of the MCMC sampling starting from the last draw of the previous run.
`S`	a positive integer, the number of posterior draws to be generated
`thin`	a positive integer, specifying the frequency of MCMC output thinning
`show_progress`	a logical value, if `TRUE` the estimation progress bar is visible

Details

The Bayesian Hierarchical Panel Vector Autoregressive model with fixed or estimated country grouping for global prior parametrs is estimated using the Gibbs sampler. In this estimation procedure all the parameters of the model are estimated jointly. The list of parameters of the model includes:

$\mathbf{A}_c$ a `KxN` country-specific autoregressive parameters matrix for each of the countries $c = 1, \dots, C$

$\mathbf{\Sigma}_c$ an `NxN` country-specific covariance matrix for each of the countries $c = 1, \dots, C$

$\mathbf{A}_g$ a `KxNxG` group-specific global autoregressive parameters matrix

$\mathbf{\Sigma}_g$ an `NxN` group-specific global covariance matrix

$\mathbf{V}$ a KxK covariance matrix of prior for global autoregressive parameters

ν prior degrees of freedom parameter

m prior average global persistence parameter

w prior scaling parameter

s prior scaling parameter

Parameters $\mathbf{A}_c$ and $\mathbf{\Sigma}_c$ have prior expected values determined by the group-specific prior parameters $\mathbf{A}_g$ and $\mathbf{\Sigma}_g$.

Gibbs sampler is an algorithm to sample random draws from the posterior distribution of the parameters of the model given the data. The algorithm is briefly explained on an example of a two-parameter model with parameters θ_1 and θ_2. In order to sample from the joint posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$ the Gibbs sampler proceeds by sampling from full-conditional posterior distributions of each parameter given data and all the other parameters, denoted by $p(\theta_1|\theta_2, \mathbf{Y})$ and $p(\theta_2|\theta_1, \mathbf{Y})$. These distributions are available from derivations and should be in a form of distributions that are easy to sample random numbers from.

To obtain S draws from the posterior distribution:

1. Set the initial values of the parameters $\theta_2^{(0)}$
2. At each of the s iterations:
   (a) Sample $\theta_1^{(s)}$ from $p(\theta_1|\theta_2^{(s-1)}, \mathbf{Y})$
   (b) Sample $\theta_2^{(s)}$ from $p(\theta_2|\theta_1^{(s)}, \mathbf{Y})$
3. Repeat step 2. S times. Return $\{\theta_1^{(s)}, \theta_2^{(s)}\}_{s=1}^{S}$ as a sample drawn from the posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$.

The `estimate()` function returns the draws from the posterior distribution of the parameters of the hierarchical panel VAR model listed above.

Thinning. Thinning is a procedure to reduce the dependence in the returned sample from the posterior distribution. It is obtained by returning every `thin` draw in the final sample. This procedure reduces the number of draws returned by the `estimate()` function.

Value

An object of class PosteriorBVARGROUPPRIORPANEL containing the Bayesian estimation output and containing two elements:

`posterior` a list with a collection of S draws from the posterior distribution generated via Gibbs sampler. Elements of the list correspond to the parameters of the model listed in section **Details** and are named respectively: A_c, Sigma_c, A_g, Sigma_g, V, nu, m, w, s.

`last_draw` an object of class BVARGROUPPRIORPANEL with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using the `estimate()` method.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

bpvars, specify_bvarGroupPriorPANEL, specify_posterior_bvarGroupPriorPANEL, summary.PosteriorBVARGROUPPRIORPANEL, forecast.PosteriorBVARGROUPPRIORPANEL

Examples

```
# specify the model
specification = specify_bvarGroupPriorPANEL$new(
     data = ilo_dynamic_panel[1:5],
     exogenous = ilo_exogenous_variables[1:5],
     group_allocation = country_grouping_region[1:5]
)
burn_in      = estimate(specification, 10)          # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 10)             # estimate the model; use say S = 10000
```

---

estimate.PosteriorBVARPANEL

Bayesian estimation of a Bayesian Hierarchical Panel Vector Autoregression using Gibbs sampler

---

Description

Estimates the Bayesian Hierarchical Panel VAR using the Gibbs sampler proposed by Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'PosteriorBVARPANEL'
estimate(specification, S, thin = 1, show_progress = TRUE)
```

Arguments

specification	an object of class PosteriorBVARPANEL generated using the estimate.BVARPANEL() function. This setup facilitates the continuation of the MCMC sampling starting from the last draw of the previous run.
S	a positive integer, the number of posterior draws to be generated
thin	a positive integer, specifying the frequency of MCMC output thinning
show_progress	a logical value, if TRUE the estimation progress bar is visible

Details

The Bayesian Hierarchical Panel Vector Autoregressive model described in bpvars is estimated using the Gibbs sampler. In this estimation procedure all the parameters of the model are estimated jointly. The list of parameters of the model includes:

$\mathbf{A}_c$ a KxN country-specific autoregressive parameters matrix for each of the countries $c = 1, \dots, C$

$\mathbf{\Sigma}_c$ an NxN country-specific covariance matrix for each of the countries $c = 1, \dots, C$

$\mathbf{A}$ a KxN global autoregressive parameters matrix

$\mathbf{\Sigma}$ an NxN global covariance matrix

$\mathbf{V}$ a KxK covariance matrix of prior for global autoregressive parameters

ν prior degrees of freedom parameter

m prior average global persistence parameter

w prior scaling parameter

s prior scaling parameter

Gibbs sampler is an algorithm to sample random draws from the posterior distribution of the parameters of the model given the data. The algorithm is briefly explained on an example of a two-parameter model with parameters θ_1 and θ_2. In order to sample from the joint posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$ the Gibbs sampler proceeds by sampling from full-conditional posterior distributions of each parameter given data and all the other parameters, denoted by $p(\theta_1|\theta_2, \mathbf{Y})$ and $p(\theta_2|\theta_1, \mathbf{Y})$. These distributions are available from derivations and should be in a form of distributions that are easy to sample random numbers from.

To obtain `S` draws from the posterior distribution:

1. Set the initial values of the parameters $\theta_2^{(0)}$
2. At each of the `s` iterations:
   (a) Sample $\theta_1^{(s)}$ from $p(\theta_1|\theta_2^{(s-1)}, \mathbf{Y})$
   (b) Sample $\theta_2^{(s)}$ from $p(\theta_2|\theta_1^{(s)}, \mathbf{Y})$
3. Repeat step 2. `S` times. Return $\{\theta_1^{(s)}, \theta_2^{(s)}\}_{s=1}^{S}$ as a sample drawn from the posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$.

The `estimate()` function returns the draws from the posterior distribution of the parameters of the hierarchical panel VAR model listed above.

Thinning. Thinning is a procedure to reduce the dependence in the returned sample from the posterior distribution. It is obtained by returning every `thin` draw in the final sample. This procedure reduces the number of draws returned by the `estimate()` function.

Value

An object of class `PosteriorBVARPANEL` containing the Bayesian estimation output and containing two elements:

`posterior` a list with a collection of `S` draws from the posterior distribution generated via Gibbs sampler. Elements of the list correspond to the parameters of the model listed in section **Details** and are named respectively: `A_c`, `Sigma_c`, `A`, `Sigma`, `V`, `nu`, `m`, `w`, `s`.

`last_draw` an object of class `BVARPANEL` with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using the `estimate()` method.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

bpvars, specify_bvarPANEL, specify_posterior_bvarPANEL, summary.PosteriorBVARPANEL, forecast.PosteriorBVARPANEL

Examples

```
# specify the model
specification = specify_bvarPANEL$new(
  ilo_dynamic_panel[1:5],
  exogenous = ilo_exogenous_variables[1:5]
)
burn_in      = estimate(specification, 10)          # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 10)              # estimate the model; use say S = 10000
```

---

estimate.PosteriorBVARs

Bayesian estimation of a Bayesian Hierarchical Vector Autoregressions for cubic data using Gibbs sampler

---

Description

Estimates the Bayesian Hierarchical VARs for cubic data using the Gibbs sampler proposed by Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'PosteriorBVARs'
estimate(specification, S, thin = 1, show_progress = TRUE)
```

Arguments

specification	an object of class PosteriorBVARs generated using the estimate.BVARs() function. This setup facilitates the continuation of the MCMC sampling starting from the last draw of the previous run.
S	a positive integer, the number of posterior draws to be generated
thin	a positive integer, specifying the frequency of MCMC output thinning
show_progress	a logical value, if TRUE the estimation progress bar is visible

Details

The Bayesian Hierarchical Vector Autoregressive models described in bpvars is estimated using the Gibbs sampler. In this estimation procedure all the parameters of the model are estimated jointly. The list of parameters of the model includes:

$\mathbf{A}_c$ a KxN country-specific autoregressive parameters matrix for each of the countries $c = 1, \dots, C$

$\mathbf{\Sigma}_c$ an NxN country-specific covariance matrix for each of the countries $c = 1, \dots, C$

ν prior degrees of freedom parameter

m prior average global persistence parameter

w prior scaling parameter

s prior scaling parameter

Gibbs sampler is an algorithm to sample random draws from the posterior distribution of the parameters of the model given the data. The algorithm is briefly explained on an example of a two-parameter model with parameters θ_1 and θ_2. In order to sample from the joint posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$ the Gibbs sampler proceeds by sampling from full-conditional posterior distributions of each parameter given data and all the other parameters, denoted by $p(\theta_1|\theta_2, \mathbf{Y})$ and $p(\theta_2|\theta_1, \mathbf{Y})$. These distributions are available from derivations and should be in a form of distributions that are easy to sample random numbers from.

To obtain S draws from the posterior distribution:

1. Set the initial values of the parameters $\theta_2^{(0)}$
2. At each of the s iterations:
   (a) Sample $\theta_1^{(s)}$ from $p(\theta_1|\theta_2^{(s-1)}, \mathbf{Y})$
   (b) Sample $\theta_2^{(s)}$ from $p(\theta_2|\theta_1^{(s)}, \mathbf{Y})$
3. Repeat step 2. S times. Return $\{\theta_1^{(s)}, \theta_2^{(s)}\}_{s=1}^{S}$ as a sample drawn from the posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$.

The estimate() function returns the draws from the posterior distribution of the parameters of the hierarchical VAR model listed above.

Thinning. Thinning is a procedure to reduce the dependence in the returned sample from the posterior distribution. It is obtained by returning every thin draw in the final sample. This procedure reduces the number of draws returned by the estimate() function.

Value

An object of class PosteriorBVARs containing the Bayesian estimation output and containing two elements:

posterior a list with a collection of S draws from the posterior distribution generated via Gibbs sampler. Elements of the list correspond to the parameters of the model listed in section **Details** and are named respectively: A_c, Sigma_c, nu, m, w, s.

last_draw an object of class BVARs with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using the estimate() method.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

bpvars, specify_bvars, specify_posterior_bvars, summary.PosteriorBVARs, forecast.PosteriorBVARs

Examples

```
# specify the model
specification = specify_bvars$new(
  ilo_dynamic_panel[1:5],
  exogenous = ilo_exogenous_variables[1:5]
)
burn_in      = estimate(specification, 10)          # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 10)              # estimate the model; use say S = 10000
```

---

forecast.PosteriorBVARGROUPPANEL

Forecasting using Hierarchical Panel Vector Autoregressions

Description

Samples from the joint predictive density of the dependent variables for all countries at forecast horizons from 1 to `horizon` specified as an argument of the function. Also implements conditional forecasting based on the provided projections for some of the variables. See Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'PosteriorBVARGROUPPANEL'
forecast(
  object,
  horizon = 1,
  exogenous_forecast = NULL,
  conditional_forecast = NULL,
  ...
)
```

Arguments

`object` posterior estimation outcome - an object of class `PosteriorBVARGROUPPANEL` obtained by running the `estimate` function.

`horizon` a positive integer, specifying the forecasting horizon.

`exogenous_forecast` not used here ATM; included for compatibility with generic `forecast`.

`conditional_forecast` a list of length `C` containing `horizon` x `N` matrices with forecasted values for selected variables. These matrices should only contain `numeric` or `NA` values. The entries with `NA` values correspond to the values that are forecasted conditionally on the realisations provided as `numeric` values.

`...` not used

Details

The package provides a range of options regarding the forecasting procedure. They are dependent on the model and forecast specifications and include Bayesian forecasting many periods ahead, conditional forecasting, and forecasting for models with exogenous variables.

One-period-ahead predictive density. The model assumptions provided in the documentation for bpvars determine the country-specific one-period ahead conditional predictive density for the unknown vector $\mathbf{y}_{c.t+1}$ given the data available at time t and the parameters of the model. It is multivariate normal with the mean $\mathbf{A}_c'\mathbf{x}_{c.t+1}$ and the covariance matrix $\mathbf{\Sigma}_c$

$$p(\mathbf{y}_{c.t+1}|\mathbf{x}_{c.t+1}, \mathbf{A}_c, \mathbf{\Sigma}_c) = N_N(\mathbf{A}_c'\mathbf{x}_{c.t+1}, \mathbf{\Sigma}_c)$$

where $\mathbf{x}_{c.t+1}$ includes the lagged values of $\mathbf{y}_{c.t+1}$, the constant term, and, potentially, exogenous variables if they were specified by the user.

Bayesian predictive density. The one-period ahead predictive density is used to sample from the joint predictive density of the unknown future values. This predictive density is defined as a joint density of $\mathbf{y}_{c.t+h}$ at horizons $h = 1, \dots, H$, where H corresponds to the value of argument `horizon`, given the data available at time t:

$$p(\mathbf{y}_{c.T_c+H}, \dots, \mathbf{y}_{c.T_c+1}|\mathbf{Y}_c, \mathbf{X}_c) = \int p(\mathbf{y}_{c.T_c+H}, \dots, \mathbf{y}_{c.T_c+1}|\mathbf{Y}_c, \mathbf{X}_c, \mathbf{A}_c, \mathbf{\Sigma}_c)p(\mathbf{A}_c, \mathbf{\Sigma}_c|\mathbf{Y}_c, \mathbf{X}_c)d(\mathbf{A}_c, \mathbf{\Sigma}_c)$$

Therefore, the Bayesian forecast does not depend on the parameter values as the parameters are integrated out with respect to their posterior distribution. Consequently, Bayesian forecasts incorporate the uncertainty with respect to estimation. Sampling from the density is facilitated using the draws from the posterior density and sequential sampling from the one-period ahead predictive density.

Conditional forecasting of some of the variables given the future values of the remaining variables is implemented following Waggoner and Zha (1999) and is based on the conditional normal density given the future projections of some of the variables created basing on the one-period ahead predictive density.

Exogenous variables. Forecasting with models for which specification argument `exogenous_variables` was specified required providing the future values of these exogenous variables in the argument `exogenous_forecast` of the `forecast.PosteriorBVARPANEL` function.

Truncated forecasts for variables of type 'rate'. The package provides the option to truncate the forecasts for variables of for which the corresponding element of argument `type` of the function `specify_bvarPANEL$new()` is set to `"rate"`. The one-period-ahead predictive normal density for such variables is truncated to values from interval $[0, 100]$.

Value

A list of class `ForecastsPANEL` with `C` elements containing the draws from the country-specific predictive density and data in a form of object class `Forecasts` that includes:

forecasts an `NxhorizonxS` array with the draws from the country-specific predictive density

forecast_mean an `NxhorizonxS` array with the mean of the country-specific predictive density

forecast_cov an `NxNxhorizonxS` array with the covariance of the country-specific predictive density

Y a `T_cxN` matrix with the country-specific data

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Waggoner, D. F., & Zha, T. (1999) Conditional forecasts in dynamic multivariate models, *Review of Economics and Statistics*, **81**(4), 639-651, doi:10.1162/003465399558508.

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`specify_bvarGroupPANEL`, `estimate.PosteriorBVARGROUPPANEL`, `summary.ForecastsPANEL`, `plot.ForecastsPANEL`

Examples

```
# specify the model
specification = specify_bvarGroupPANEL$new(
                  ilo_dynamic_panel[1:5],
                  exogenous = ilo_exogenous_variables[1:5],
                  group_allocation = country_grouping_incomegroup[1:5]
                )
burn_in      = estimate(specification, 5)           # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 5)               # estimate the model; use say S = 10000

# forecast 3 years ahead
predictive    = forecast(
                  posterior,
                  horizon = 3,
                  exogenous_forecast = ilo_exogenous_forecasts[1:5]
                )
```

`forecast.PosteriorBVARGROUPPRIORPANEL`
Forecasting using Hierarchical Panel Vector Autoregressions

Description

Samples from the joint predictive density of the dependent variables for all countries at forecast horizons from 1 to `horizon` specified as an argument of the function. Also implements conditional forecasting based on the provided projections for some of the variables. See Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'PosteriorBVARGROUPPRIORPANEL'
forecast(
  object,
  horizon = 1,
  exogenous_forecast = NULL,
  conditional_forecast = NULL,
  ...
)
```

Arguments

- `object` posterior estimation outcome - an object of class `PosteriorBVARGROUPPRIORPANEL` obtained by running the `estimate` function.
- `horizon` a positive integer, specifying the forecasting horizon.
- `exogenous_forecast` not used here ATM; included for compatibility with generic `forecast`.
- `conditional_forecast` a list of length `C` containing `horizon` `x` `N` matrices with forecasted values for selected variables. These matrices should only contain `numeric` or `NA` values. The entries with `NA` values correspond to the values that are forecasted conditionally on the realisations provided as `numeric` values.
- `...` not used

Details

The package provides a range of options regarding the forecasting procedure. They are dependent on the model and forecast specifications and include Bayesian forecasting many periods ahead, conditional forecasting, and forecasting for models with exogenous variables.

One-period-ahead predictive density. The model assumptions provided in the documentation for bpvars determine the country-specific one-period ahead conditional predictive density for the unknown vector $\mathbf{y}_{c.t+1}$ given the data available at time t and the parameters of the model. It is multivariate normal with the mean $\mathbf{A}_c'\mathbf{x}_{c.t+1}$ and the covariance matrix $\mathbf{\Sigma}_c$

$$p(\mathbf{y}_{c.t+1}|\mathbf{x}_{c.t+1}, \mathbf{A}_c, \mathbf{\Sigma}_c) = N_N(\mathbf{A}_c'\mathbf{x}_{c.t+1}, \mathbf{\Sigma}_c)$$

where $\mathbf{x}_{c.t+1}$ includes the lagged values of $\mathbf{y}_{c.t+1}$, the constant term, and, potentially, exogenous variables if they were specified by the user.

Bayesian predictive density. The one-period ahead predictive density is used to sample from the joint predictive density of the unknown future values. This predictive density is defined as a joint density of $\mathbf{y}_{c.t+h}$ at horizons $h = 1, \dots, H$, where H corresponds to the value of argument `horizon`, given the data available at time t:

$$p(\mathbf{y}_{c.T_c+H}, \dots, \mathbf{y}_{c.T_c+1} | \mathbf{Y}_c, \mathbf{X}_c) = \int p(\mathbf{y}_{c.T_c+H}, \dots, \mathbf{y}_{c.T_c+1} | \mathbf{Y}_c, \mathbf{X}_c, \mathbf{A}_c, \mathbf{\Sigma}_c) p(\mathbf{A}_c, \mathbf{\Sigma}_c | \mathbf{Y}_c, \mathbf{X}_c) d(\mathbf{A}_c, \mathbf{\Sigma}_c)$$

Therefore, the Bayesian forecast does not depend on the parameter values as the parameters are integrated out with respect to their posterior distribution. Consequently, Bayesian forecasts incorporate the uncertainty with respect to estimation. Sampling from the density is facilitated using the draws from the posterior density and sequential sampling from the one-period ahead predictive density.

Conditional forecasting of some of the variables given the future values of the remaining variables is implemented following Waggoner and Zha (1999) and is based on the conditional normal density given the future projections of some of the variables created basing on the one-period ahead predictive density.

Exogenous variables. Forecasting with models for which specification argument `exogenous_variables` was specified required providing the future values of these exogenous variables in the argument `exogenous_forecast` of the `forecast.PosteriorBVARPANEL` function.

Truncated forecasts for variables of type 'rate'. The package provides the option to truncate the forecasts for variables of for which the corresponding element of argument `type` of the function `specify_bvarPANEL$new()` is set to `"rate"`. The one-period-ahead predictive normal density for such variables is truncated to values from interval $[0, 100]$.

Value

A list of class `ForecastsPANEL` with `C` elements containing the draws from the country-specific predictive density and data in a form of object class `Forecasts` that includes:

forecasts an `NxhorizonxS` array with the draws from the country-specific predictive density

forecast_mean an `NxhorizonxS` array with the mean of the country-specific predictive density

forecast_cov an `NxNxhorizonxS` array with the covariance of the country-specific predictive density

Y a `T_cxN` matrix with the country-specific data

Author(s)

Tomasz Woźniak `<wozniak.tom@pm.me>`

References

Waggoner, D. F., & Zha, T. (1999) Conditional forecasts in dynamic multivariate models, *Review of Economics and Statistics*, **81**(4), 639-651, doi:10.1162/003465399558508.

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

specify_bvarGroupPriorPANEL, estimate.PosteriorBVARGROUPPRIORPANEL, summary.ForecastsPANEL, plot.ForecastsPANEL

Examples

```
# specify the model
specification = specify_bvarGroupPriorPANEL$new(
                  ilo_dynamic_panel[1:5],
                  group_allocation = country_grouping_incomegroup[1:5]
                )
burn_in      = estimate(specification, 5)           # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 5)               # estimate the model; use say S = 10000

# forecast 3 years ahead
predictive    = forecast(posterior, horizon = 3)
```

---

forecast.PosteriorBVARPANEL
Forecasting using Hierarchical Panel Vector Autoregressions

Description

Samples from the joint predictive density of the dependent variables for all countries at forecast horizons from 1 to `horizon` specified as an argument of the function. Also implements conditional forecasting based on the provided projections for some of the variables. See Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'PosteriorBVARPANEL'
forecast(
  object,
  horizon = 1,
  exogenous_forecast = NULL,
  conditional_forecast = NULL,
  ...
)
```

Arguments

`object`	posterior estimation outcome - an object of class `PosteriorBVARPANEL` obtained by running the `estimate` function.
`horizon`	a positive integer, specifying the forecasting horizon.
`exogenous_forecast`	not used here ATM; included for compatibility with generic `forecast`.

`conditional_forecast`
: a list of length `C` containing `horizon` x `N` matrices with forecasted values for selected variables. These matrices should only contain `numeric` or `NA` values. The entries with `NA` values correspond to the values that are forecasted conditionally on the realisations provided as `numeric` values.

`...`
: not used

Details

The package provides a range of options regarding the forecasting procedure. They are dependent on the model and forecast specifications and include Bayesian forecasting many periods ahead, conditional forecasting, and forecasting for models with exogenous variables.

One-period-ahead predictive density. The model assumptions provided in the documentation for `bpvars` determine the country-specific one-period ahead conditional predictive density for the unknown vector $\mathbf{y}_{c.t+1}$ given the data available at time t and the parameters of the model. It is multivariate normal with the mean $\mathbf{A}_c'\mathbf{x}_{c.t+1}$ and the covariance matrix $\mathbf{\Sigma}_c$

$$p(\mathbf{y}_{c.t+1}|\mathbf{x}_{c.t+1}, \mathbf{A}_c, \mathbf{\Sigma}_c) = N_N(\mathbf{A}_c'\mathbf{x}_{c.t+1}, \mathbf{\Sigma}_c)$$

where $\mathbf{x}_{c.t+1}$ includes the lagged values of $\mathbf{y}_{c.t+1}$, the constant term, and, potentially, exogenous variables if they were specified by the user.

Bayesian predictive density. The one-period ahead predictive density is used to sample from the joint predictive density of the unknown future values. This predictive density is defined as a joint density of $\mathbf{y}_{c.t+h}$ at horizons $h = 1, \dots, H$, where H corresponds to the value of argument `horizon`, given the data available at time t:

$$p(\mathbf{y}_{c.T_c+H}, \dots, \mathbf{y}_{c.T_c+1}|\mathbf{Y}_c, \mathbf{X}_c) = \int p(\mathbf{y}_{c.T_c+H}, \dots, \mathbf{y}_{c.T_c+1}|\mathbf{Y}_c, \mathbf{X}_c, \mathbf{A}_c, \mathbf{\Sigma}_c)p(\mathbf{A}_c, \mathbf{\Sigma}_c|\mathbf{Y}_c, \mathbf{X}_c)d(\mathbf{A}_c, \mathbf{\Sigma}_c)$$

Therefore, the Bayesian forecast does not depend on the parameter values as the parameters are integrated out with respect to their posterior distribution. Consequently, Bayesian forecasts incorporate the uncertainty with respect to estimation. Sampling from the density is facilitated using the draws from the posterior density and sequential sampling from the one-period ahead predictive density.

Conditional forecasting of some of the variables given the future values of the remaining variables is implemented following Waggoner and Zha (1999) and is based on the conditional normal density given the future projections of some of the variables created basing on the one-period ahead predictive density.

Exogenous variables. Forecasting with models for which specification argument `exogenous_variables` was specified required providing the future values of these exogenous variables in the argument `exogenous_forecast` of the `forecast.PosteriorBVARPANEL` function.

Truncated forecasts for variables of type 'rate'. The package provides the option to truncate the forecasts for variables of for which the corresponding element of argument `type` of the function `specify_bvarPANEL$new()` is set to `"rate"`. The one-period-ahead predictive normal density for such variables is truncated to values from interval $[0, 100]$.

Value

A list of class ForecastsPANEL with C elements containing the draws from the country-specific predictive density and data in a form of object class Forecasts that includes:

forecasts an NxhorizonxS array with the draws from the country-specific predictive density

forecast_mean an NxhorizonxS array with the mean of the country-specific predictive density

forecast_cov an NxNxhorizonxS array with the covariance of the country-specific predictive density

Y a T_cxN matrix with the country-specific data

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Waggoner, D. F., & Zha, T. (1999) Conditional forecasts in dynamic multivariate models, *Review of Economics and Statistics*, **81**(4), 639-651, doi:10.1162/003465399558508.

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

specify_bvarPANEL, estimate.PosteriorBVARPANEL, summary.ForecastsPANEL, plot.ForecastsPANEL

Examples

```
# specify the model
specification = specify_bvarPANEL$new(
  ilo_dynamic_panel[1:5],
  exogenous = ilo_exogenous_variables[1:5]
)
burn_in      = estimate(specification, 5)           # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 5)              # estimate the model; use say S = 10000

# forecast 3 years ahead
predictive    = forecast(posterior, 3, exogenous_forecast = ilo_exogenous_forecasts[1:5])
```

---

forecast.PosteriorBVARs

Forecasting using Hierarchical Vector Autoregressions for Dynamic Panel Data

---

Description

Samples from the joint predictive density of the dependent variables for all countries at forecast horizons from 1 to `horizon` specified as an argument of the function. Also implements conditional forecasting based on the provided projections for some of the variables. See Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'PosteriorBVARs'
forecast(
  object,
  horizon = 1,
  exogenous_forecast = NULL,
  conditional_forecast = NULL,
  ...
)
```

Arguments

`object`	posterior estimation outcome - an object of class `PosteriorBVARs` obtained by running the `estimate` function.
`horizon`	a positive integer, specifying the forecasting horizon.
`exogenous_forecast`	not used here ATM; included for compatibility with generic `forecast`.
`conditional_forecast`	a list of length `C` containing `horizon x N` matrices with forecasted values for selected variables. These matrices should only contain `numeric` or `NA` values. The entries with `NA` values correspond to the values that are forecasted conditionally on the realisations provided as `numeric` values.
`...`	not used

Details

The package provides a range of options regarding the forecasting procedure. They are dependent on the model and forecast specifications and include Bayesian forecasting many periods ahead, conditional forecasting, and forecasting for models with exogenous variables.

One-period-ahead predictive density. The model assumptions provided in the documentation for `bpvars` determine the country-specific one-period ahead conditional predictive density for the unknown vector $\mathbf{y}_{c.t+1}$ given the data available at time t and the parameters of the model. It is multivariate normal with the mean $\mathbf{A}_c'\mathbf{x}_{c.t+1}$ and the covariance matrix $\mathbf{\Sigma}_c$

$$p(\mathbf{y}_{c.t+1}|\mathbf{x}_{c.t+1}, \mathbf{A}_c, \mathbf{\Sigma}_c) = N_N(\mathbf{A}_c'\mathbf{x}_{c.t+1}, \mathbf{\Sigma}_c)$$

where $\mathbf{x}_{c.t+1}$ includes the lagged values of $\mathbf{y}_{c.t+1}$, the constant term, and, potentially, exogenous variables if they were specified by the user.

Bayesian predictive density. The one-period ahead predictive density is used to sample from the joint predictive density of the unknown future values. This predictive density is defined as a joint density of $\mathbf{y}_{c.t+h}$ at horizons $h = 1, \dots, H$, where H corresponds to the value of argument `horizon`, given the data available at time t:

$$p(\mathbf{y}_{c.T_c+H}, \dots, \mathbf{y}_{c.T_c+1}|\mathbf{Y}_c, \mathbf{X}_c) = \int p(\mathbf{y}_{c.T_c+H}, \dots, \mathbf{y}_{c.T_c+1}|\mathbf{Y}_c, \mathbf{X}_c, \mathbf{A}_c, \mathbf{\Sigma}_c)p(\mathbf{A}_c, \mathbf{\Sigma}_c|\mathbf{Y}_c, \mathbf{X}_c)d(\mathbf{A}_c, \mathbf{\Sigma}_c)$$

Therefore, the Bayesian forecast does not depend on the parameter values as the parameters are integrated out with respect to their posterior distribution. Consequently, Bayesian forecasts incorporate the uncertainty with respect to estimation. Sampling from the density is facilitated using the draws from the posterior density and sequential sampling from the one-period ahead predictive density.

Conditional forecasting of some of the variables given the future values of the remaining variables is implemented following Waggoner and Zha (1999) and is based on the conditional normal density given the future projections of some of the variables created basing on the one-period ahead predictive density.

Exogenous variables. Forecasting with models for which specification argument `exogenous_variables` was specified required providing the future values of these exogenous variables in the argument `exogenous_forecast` of the `forecast.PosteriorBVARPANEL` function.

Truncated forecasts for variables of type 'rate'. The package provides the option to truncate the forecasts for variables of for which the corresponding element of argument `type` of the function `specify_bvarPANEL$new()` is set to `"rate"`. The one-period-ahead predictive normal density for such variables is truncated to values from interval $[0, 100]$.

Value

A list of class `ForecastsPANEL` with `C` elements containing the draws from the country-specific predictive density and data in a form of object class `Forecasts` that includes:

forecasts an `NxhorizonxS` array with the draws from the country-specific predictive density

forecast_mean an `NxhorizonxS` array with the mean of the country-specific predictive density

forecast_cov an `NxNxhorizonxS` array with the covariance of the country-specific predictive density

Y a `T_cxN` matrix with the country-specific data

Author(s)

Tomasz Woźniak `<wozniak.tom@pm.me>`

References

Waggoner, D. F., & Zha, T. (1999) Conditional forecasts in dynamic multivariate models, *Review of Economics and Statistics*, **81**(4), 639-651, doi:10.1162/003465399558508.

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`specify_bvars`, `estimate.PosteriorBVARs`, `summary.ForecastsPANEL`, `plot.ForecastsPANEL`

Examples

```
# specify the model
specification = specify_bvars$new(
  ilo_dynamic_panel[1:5],
  exogenous = ilo_exogenous_variables[1:5]
)
burn_in       = estimate(specification, 5)           # run the burn-in; use say S = 10000
posterior     = estimate(burn_in, 5)               # estimate the model; use say S = 10000

# forecast 3 years ahead
predictive    = forecast(posterior, 3, exogenous_forecast = ilo_exogenous_forecasts[1:5])
```

---

forecast_poos_recursively

Bayesian recursive pseudo-out-of-sample forecasting

Description

Performs the recursive pseudo-out-of-sample forecasting exercise using expanding window samples following Sanchez-Martinez & Woźniak (2026).

Usage

```
forecast_poos_recursively(model_spec, poos_spec, show_progress = TRUE)
```

Arguments

`model_spec`	an object generated using one of the `specify_*` functions containing model specification.
`poos_spec`	an object of class `POOSForecastSetup` containing specification of the recursive pseudo-out-of-sample forecasting exercise using expanding window samples.
`show_progress`	a logical value, if `TRUE` the estimation progress bar is visible

Value

An object of class `ForecastsPOOS` containing the outcome of Bayesian recursive pseudo-out-of-sample forecasting exercise using expanding window samples. The object is a list with `forecasting_sample` elements, where `forecasting_sample` is equal to the sample size less the maximum of `horizons` and the `training_sample` plus one. Each element of the list is an object of class `ForecastsPANEL` containing the forecasts for each country, see `forecast.PosteriorBVARPANEL`.

Author(s)

Tomasz Woźniak <`wozniak.tom@pm.me`>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

forecast.PosteriorBVARPANEL, specify_bvarPANEL, specify_poosf_exercise, estimate.BVARPANEL

Examples

```
spec = specify_bvarPANEL$new(ilo_dynamic_panel[1:5])   # specify the model
poos = specify_poosf_exercise$new(                     # specify the forecasting exercise
         spec,
         S = 5,                                         # use at least S = 5000
         S_burn = 2,                                    # use at least S_burn = 1000
         horizons = 1,
         training_sample = 30
       )
fore = forecast_poos_recursively(spec, poos)          # execute the forecasting exercise
```

---

forecast_poos_recursively.BVARGROUPPANEL

Bayesian recursive pseudo-out-of-sample forecasting

Description

Performs the recursive pseudo-out-of-sample forecasting exercise using expanding window samples following Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'BVARGROUPPANEL'
forecast_poos_recursively(model_spec, poos_spec, show_progress = TRUE)
```

Arguments

model_spec	an object of class BVARGROUPPANEL generated using the specify_bvarGroupPANEL function and containing the Bayesian Panel VAR model specification.
poos_spec	an object of class POOSForecastSetup containing specification of the recursive pseudo-out-of-sample forecasting exercise using expanding window samples.
show_progress	a logical value, if TRUE the estimation progress bar is visible

Value

An object of class ForecastsPOOS containing the outcome of Bayesian recursive pseudo-out-of-sample forecasting exercise using expanding window samples. The object is a list with forecasting_sample elements, where forecasting_sample is equal to the sample size less the maximum of horizons and the training_sample plus one. Each element of the list is an object of class ForecastsPANEL containing the forecasts for each country, see forecast.PosteriorBVARPANEL.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

forecast.PosteriorBVARPANEL, specify_bvarPANEL, specify_poosf_exercise, estimate.BVARPANEL

Examples

```
spec = specify_bvarGroupPANEL$new(                # specify the model
         ilo_dynamic_panel[1:5],
         group_allocation = country_grouping_region[1:5]
       )
poos = specify_poosf_exercise$new(                # specify the forecasting exercise
         spec,
         S = 5,                                  # use at least S = 5000
         S_burn = 2,                              # use at least S_burn = 1000
         horizons = 1,
         training_sample = 30
       )
fore = forecast_poos_recursively(spec, poos)      # execute the forecasting exercise
```

---

forecast_poos_recursively.BVARGROUPPRIORPANEL
Bayesian recursive pseudo-out-of-sample forecasting

Description

Performs the recursive pseudo-out-of-sample forecasting exercise using expanding window samples following Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'BVARGROUPPRIORPANEL'
forecast_poos_recursively(model_spec, poos_spec, show_progress = TRUE)
```

Arguments

model_spec — an object of class BVARGROUPPRIORPANEL generated using the specify_bvarGroupPANEL function and containing the Bayesian Panel VAR model specification with group-specific global parameters.

poos_spec — an object of class POOSForecastSetup containing specification of the recursive pseudo-out-of-sample forecasting exercise using expanding window samples.

show_progress — a logical value, if TRUE the estimation progress bar is visible

Value

An object of class ForecastsPOOS containing the outcome of Bayesian recursive pseudo-out-of-sample forecasting exercise using expanding window samples. The object is a list with forecasting_sample elements, where forecasting_sample is equal to the sample size less the maximum of horizons and the training_sample plus one. Each element of the list is an object of class ForecastsPANEL containing the forecasts for each country, see forecast.PosteriorBVARPANEL.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

forecast.PosteriorBVARPANEL, specify_bvarPANEL, specify_poosf_exercise, estimate.BVARPANEL

Examples

```
spec = specify_bvarGroupPriorPANEL$new(                # specify the model
         ilo_dynamic_panel[1:5],
         group_allocation = country_grouping_region[1:5]
       )
poos = specify_poosf_exercise$new(                # specify the forecasting exercise
         spec,
         S = 5,                                  # use at least S = 5000
         S_burn = 5,                              # use at least S_burn = 1000
         horizons = 1,
         training_sample = 30
       )
fore = forecast_poos_recursively(spec, poos)      # execute the forecasting exercise
```

`forecast_poos_recursively.BVARPANEL`
Bayesian recursive pseudo-out-of-sample forecasting

Description

Performs the recursive pseudo-out-of-sample forecasting exercise using expanding window samples following Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'BVARPANEL'
forecast_poos_recursively(model_spec, poos_spec, show_progress = TRUE)
```

Arguments

`model_spec`	an object of class `BVARPANEL` generated using the `specify_bvarPANEL` function and containing the Bayesian Panel VAR model specification.
`poos_spec`	an object of class `POOSForecastSetup` containing specification of the recursive pseudo-out-of-sample forecasting exercise using expanding window samples.
`show_progress`	a logical value, if `TRUE` the estimation progress bar is visible

Value

An object of class `ForecastsPOOS` containing the outcome of Bayesian recursive pseudo-out-of-sample forecasting exercise using expanding window samples. The object is a list with `forecasting_sample` elements, where `forecasting_sample` is equal to the sample size less the maximum of `horizons` and the `training_sample` plus one. Each element of the list is an object of class `ForecastsPANEL` containing the forecasts for each country, see `forecast.PosteriorBVARPANEL`.

Author(s)

Tomasz Woźniak `<wozniak.tom@pm.me>`

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`forecast.PosteriorBVARPANEL`, `specify_bvarPANEL`, `specify_poosf_exercise`, `estimate.BVARPANEL`

Examples

```
spec = specify_bvarPANEL$new(ilo_dynamic_panel[1:5])   # specify the model
poos = specify_poosf_exercise$new(                    # specify the forecasting exercise
         spec,
         S = 5,                                       # use at least S = 5000
         S_burn = 2,                                  # use at least S_burn = 1000
         horizons = 1,
         training_sample = 30
       )
fore = forecast_poos_recursively(spec, poos)         # execute the forecasting exercise
```

`forecast_poos_recursively.BVARs`
Bayesian recursive pseudo-out-of-sample forecasting

Description

Performs the recursive pseudo-out-of-sample forecasting exercise using expanding window samples following Sanchez-Martinez & Woźniak (2026).

Usage

```
## S3 method for class 'BVARs'
forecast_poos_recursively(model_spec, poos_spec, show_progress = TRUE)
```

Arguments

`model_spec`	an object of class `BVARs` generated using the `specify_bvars` function and containing the Bayesian VAR models specification.
`poos_spec`	an object of class `POOSForecastSetup` containing specification of the recursive pseudo-out-of-sample forecasting exercise using expanding window samples.
`show_progress`	a logical value, if `TRUE` the estimation progress bar is visible

Value

An object of class `ForecastsPOOS` containing the outcome of Bayesian recursive pseudo-out-of-sample forecasting exercise using expanding window samples. The object is a list with `forecasting_sample` elements, where `forecasting_sample` is equal to the sample size less the maximum of `horizons` and the `training_sample` plus one. Each element of the list is an object of class `ForecastsPANEL` containing the forecasts for each country, see `forecast.PosteriorBVARPANEL`.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Sanchez-Martinez & Woźniak (2026). Forecasting with Bayesian Panel Vector Autoregressions Using the R Package bpvars. University of Melbourne Working Paper, 1-39, doi:10.48550/arXiv.2606.14143.

See Also

`forecast.PosteriorBVARPANEL`, `specify_bvarPANEL`, `specify_poosf_exercise`, `estimate.BVARPANEL`

Examples

```
spec = specify_bvars$new(ilo_dynamic_panel[1:5])  # specify the model
poos = specify_poosf_exercise$new(                # specify the forecasting exercise
         spec,
         S = 5,                                  # use at least S = 5000
         S_burn = 5,                              # use at least S_burn = 1000
         horizons = 1,
         training_sample = 30
       )
fore = forecast_poos_recursively(spec, poos)      # execute the forecasting exercise
```

---

ilo_conditional_forecasts

Data containing future observations for 189 countries from 2025 to 2027 to be used for conditional forecasts given the future values of gdp.

Description

For each of the countries a time series of 3 observations on the future values of gdp is provided. These future values are taken from IMF projections. The future values of other variables are set to `NA`. Last data update was implemented on 2026-04-18.

Usage

```
data(ilo_conditional_forecasts)
```

Format

A list of 189 `ts` objects with time series of 3 observations on 4 variables:

gdp logarithm of gross domestic product

UR annual unemployment rate

EPR annual employment rate

LFPR annual labour force participation rate

Examples

```
data(ilo_conditional_forecasts)   # upload the data
```

`ilo_dynamic_panel` *A 4-variable annual system for forecasting labour market outcomes for 189 countries from 1991 to 2024*

Description

For each of the countries a time series of 34 observations on 4 variables including the logarithm of Gross Domestic Product (gdp), as well as the labour market outcomes including the unemployment rate (UR), employment rate (EPR), labour force participation rate (LFPR). The missing observations are filled using imputation method. Last data update was implemented on 2026-04-18.

Usage

```
data(ilo_dynamic_panel)
```

Format

A list of 189 `ts` objects with time series of 34 observations on 4 variables:

gdp logarithm of gross domestic product

UR annual unemployment rate

EPR annual employment rate

LFPR annual labour force participation rate

Source

International Labour Organization. (2020). ILO modelled estimates database, ILOSTAT [database]. Available from https://ilostat.ilo.org/data/.

Examples

```
data(ilo_dynamic_panel)   # upload the data
```

`ilo_dynamic_panel_missing`

A 4-variable annual system for forecasting labour market outcomes for 189 countries to 2024 containing only actual observations

---

Description

For each of the countries a time series of observations on 4 variables including the logarithm of Gross Domestic Product (gdp), as well as the labour market outcomes including the unemployment rate (UR), employment rate (EPR), labour force participation rate (LFPR). The series are of various lengths and contain missing values due to data availability. Last data update was implemented on 2026-04-20.

Usage

```
data(ilo_dynamic_panel_missing)
```

Format

A list of 189 `ts` objects with time series on 4 variables:

gdp logarithm of gross domestic product

UR annual unemployment rate

EPR annual employment rate

LFPR annual labour force participation rate

Source

International Labour Organization. (2020). ILO modelled estimates database, ILOSTAT [database]. Available from https://ilostat.ilo.org/data/.

Examples

```
data(ilo_dynamic_panel_missing)   # upload the data
```

`ilo_exogenous_forecasts`

Data containing future observations for 189 countries from 2025 to 2027 to be used to forecast with models with `ilo_exogenous_variables`

Description

For each of the countries a time series of 3 observations on On the dummies is provided. These future values are all equal to zero. They provide benchmark for the objects to be used when `exogenous_variables` are used. Last data update was implemented on 2026-04-18.

Usage

```
data(ilo_exogenous_forecasts)
```

Format

A list of 189 `ts` objects with time series of 3 observations on 3 variables:

2008 the aftermath of the Global Financial Crisis

2020 the COVID pandemic

2021 the aftermath of the COVID pandemic

Examples

```
data(ilo_exogenous_forecasts)   # upload the data
```

`ilo_exogenous_variables` *A 3-variable annual system for of dummy observations for 2008, 2020, and 2021 to be used in the estimation of the Panel VAR model for 189 countries from 1991 to 2024*

Description

For each of the countries a time series of 33 observations on 3 dummy variables for the years 2008, 2020, and 2021 is provided. Last data update was implemented on 2026-04-18.

Usage

```
data(ilo_exogenous_variables)
```

Format

A list of 189 `ts` objects with time series of 34 observations on 3 variables:

2008 the aftermath of the Global Financial Crisis

2020 the COVID pandemic

2021 the aftermath of the COVID pandemic

Examples

```
data(ilo_exogenous_variables)   # upload the data
```

`ilo_exogenous_variables_missing`
A 3-variable annual system for of dummy observations for 2008, 2020, and 2021 to be used in the estimation of the Panel VAR model for 189 countries to 2024 containing observations for matching periods from `ilo_dynamic_panel_missing`

Description

For each of the countries a time series of observations on 3 dummy variables for the years 2008, 2020, and 2021 is provided. Last data update was implemented on 2026-04-20.

Usage

```
data(ilo_exogenous_variables_missing)
```

Format

A list of 189 `ts` objects with time series on 3 variables:

2008 the aftermath of the Global Financial Crisis

2020 the COVID pandemic

2021 the aftermath of the COVID pandemic

Examples

```
data(ilo_exogenous_variables_missing)   # upload the data
```

`plot.ForecastsPANEL` *Plots fitted values of dependent variables*

Description

Plots of fitted values of dependent variables including their median and percentiles.

Usage

```
## S3 method for class 'ForecastsPANEL'
plot(
  x,
  which_c,
  probability = 0.9,
  data_in_plot = 1,
  col = "#1614B1",
```

```
  main,
  xlab,
  mar.multi = c(1, 4.6, 0, 2.1),
  oma.multi = c(6, 0, 5, 0),
  ...
)
```

Arguments

`x`	an object of class `ForecastsPANEL` obtained using the `forecast()` function containing posterior draws of fitted values of dependent variables.
`which_c`	a positive integer or a character string specifying the country for which the forecast should be plotted.
`probability`	a parameter determining the interval to be plotted. The interval stretches from the `0.5 * (1 - probability)` to `1 - 0.5 * (1 - probability)` percentile of the posterior distribution.
`data_in_plot`	a fraction value in the range (0, 1) determining how many of the last observations in the data should be plotted with the forecasts.
`col`	a colour of the plot line and the ribbon
`main`	an alternative main title for the plot
`xlab`	an alternative x-axis label for the plot
`mar.multi`	the default `mar` argument setting in `graphics::par`. Modify with care!
`oma.multi`	the default `oma` argument setting in `graphics::par`. Modify with care!
`...`	additional arguments affecting the summary produced.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

See Also

`forecast.PosteriorBVARPANEL`

Examples

```
specification = specify_bvarPANEL$new(ilo_dynamic_panel[1:5])  # specify the model
burn_in       = estimate(specification, 10)                    # run the burn-in
posterior     = estimate(burn_in, 10)                          # estimate the model

# forecast 6 years ahead
predictive    = forecast(posterior, 6)
plot(predictive, which_c = "ARG")                                    # plot forecasts
```

plot.PosteriorFEVDPANEL *Plots forecast error variance decompositions*

Description

Plots of the posterior means of the forecast error variance decompositions.

Usage

```
## S3 method for class 'PosteriorFEVDPANEL'
plot(
  x,
  which_c,
  cols,
  main,
  xlab,
  mar.multi = c(1, 4.6, 0, 4.6),
  oma.multi = c(6, 0, 5, 0),
  ...
)
```

Arguments

`x`	an object of class `PosteriorFEVDPANEL` obtained using the `compute_variance_decompositions()` function containing posterior draws of forecast error variance decompositions.
`which_c`	a positive integer or a character string specifying the country for which the forecast should be plotted.
`cols`	an `N`-vector with colours of the plot
`main`	an alternative main title for the plot
`xlab`	an alternative x-axis label for the plot
`mar.multi`	the default `mar` argument setting in `graphics::par`. Modify with care!
`oma.multi`	the default `oma` argument setting in `graphics::par`. Modify with care!
`...`	additional arguments affecting the summary produced.

Author(s)

Tomasz Woźniak <`wozniak.tom@pm.me`>

set.seed(123) specification = specify_bvarPANEL$new(ilo_dynamic_panel[1:5])

run the burn-in burn_in = estimate(specification, 10)

estimate the model posterior = estimate(burn_in, 20)

compute forecast error variance decomposition 4 years ahead fevd = compute_variance_decompositions(posterior, horizon = 4) plot(fevd, which_c = "ARG")

See Also

compute_variance_decompositions.PosteriorBVARPANEL

---

specify_bvarGroupPANEL
R6 Class representing the specification of the BVARGROUPPANEL model

---

Description

The class BVARGROUPPANEL presents complete specification for the Bayesian Panel Vector Autoregression with county groups. The groups can be pre-specified, which requires the argument `group_allocation` to be provided, or estimated, which requires the argument `G` for the number of groups to be provided and the argument `group_allocation` to be left empty.

Super class

`BVARPANEL` -> `BVARGROUPPANEL`

Public fields

`p` a non-negative integer specifying the autoregressive lag order of the model.

`G` a non-negative integer specifying the number of country groupings.

`estimate_groups` a logical value denoting whether the groups are to be estimated.

`prior` an object PriorBSVAR with the prior specification.

`data_matrices` an object DataMatricesBVARPANEL with the data matrices.

`starting_values` an object StartingValuesBVARGROUPPANEL with the starting values.

`adaptiveMH` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Methods

Public methods:

- `BVARGROUPPANEL$new()`
- `BVARGROUPPANEL$set_global2pooled()`
- `BVARGROUPPANEL$clone()`

`BVARGROUPPANEL$new()`: Create a new specification of the Bayesian Panel VAR model with country grouping BVARGROUPPANEL. The groups can be pre-specified, which requires the argument `group_allocation` to be provided, or estimated, which requires the argument `G` for the number of groups to be provided and the argument `group_allocation` to be left empty.

Usage:

```
BVARGROUPPANEL$new(
  data,
  p = 1L,
  exogenous = NULL,
  stationary = rep(FALSE, ncol(data[[1]])),
  type = rep("real", ncol(data[[1]])),
  G = NULL,
  group_allocation = NULL
)
```

Arguments:

`data` a list with `C` elements of `(T_c+p)xN` matrices with time series data.

`p` a positive integer providing model's autoregressive lag order.

`exogenous` a `(T+p)xd` matrix of exogenous variables.

`stationary` an `N` logical vector - its element set to `FALSE` sets the prior mean for the autoregressive parameters of the `N`th equation to the white noise process, otherwise to random walk.

`type` an `N` character vector with elements set to "rate" or "real" determining the truncation of the predictive density to `[0, 100]` and `(-Inf, Inf)` (no truncation) for each of the variables.

`G` a positive integer specifying the number of country groups. Its specification is required if `group_allocation` is not provided and the country groups to be estimated.

`group_allocation` an argument that can be provided as a numeric vector with integer numbers denoting group allocations to pre-specify the the country groups, in which case they are not estimated, or left empty if the country groups are to be estimated.

Returns: A new complete specification for the Bayesian Panel VAR model BVARPANEL.

`BVARGROUPPANEL$set_global2pooled()`: Sets the prior mean of the global autoregressive parameters to the OLS pooled panel estimator following Zellner, Hong (1989).

Usage:

```
BVARGROUPPANEL$set_global2pooled(x)
```

Arguments:

`x` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Examples:

```
spec = specify_bvarGroupPANEL$new(
   data = ilo_dynamic_panel[1:5],
   G = 2
)
spec$set_global2pooled()
```

`BVARGROUPPANEL$clone()`: The objects of this class are cloneable with this method.

Usage:

```
BVARGROUPPANEL$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

References

Zellner, Hong (1989). Forecasting international growth rates using Bayesian shrinkage and other procedures. *Journal of Econometrics*, **40**(1), 183–202, doi:10.1016/03044076(89)900365.

Examples

```
spec = specify_bvarGroupPANEL$new(
   data = ilo_dynamic_panel[1:5],
   G = 2
)


## ------------------------------------------------
## Method `BVARGROUPPANEL$set_global2pooled()`
## ------------------------------------------------

spec = specify_bvarGroupPANEL$new(
   data = ilo_dynamic_panel[1:5],
   G = 2
)
spec$set_global2pooled()
```

---

specify_bvarGroupPriorPANEL   *R6 Class representing the specification of the BVARGROUPPRIORPANEL model*

Description

The class BVARGROUPPRIORPANEL presents complete specification for the Bayesian Panel Vector Autoregression with county grouping for global prior parameters. The groups can be prespecified, which requires the argument `group_allocation` to be provided, or estimated, which requires the argument `G` for the number of groups to be provided and the argument `group_allocation` to be left empty.

Public fields

`p` a non-negative integer specifying the autoregressive lag order of the model.

`G` a non-negative integer specifying the number of country groupings.

`estimate_groups` a logical value denoting whether the groups are to be estimated.

`prior` an object PriorBSVAR with the prior specification.

`data_matrices` an object DataMatricesBVARPANEL with the data matrices.

`starting_values` an object StartingValuesBVARGROUPPRIORPANEL with the starting values.

`adaptiveMH` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Methods

Public methods:

- BVARGROUPPRIORPANEL$new()
- BVARGROUPPRIORPANEL$get_data_matrices()
- BVARGROUPPRIORPANEL$get_prior()
- BVARGROUPPRIORPANEL$get_starting_values()
- BVARGROUPPRIORPANEL$get_type()
- BVARGROUPPRIORPANEL$set_global2pooled()
- BVARGROUPPRIORPANEL$set_adaptiveMH()
- BVARGROUPPRIORPANEL$clone()

`BVARGROUPPRIORPANEL$new()`: Create a new specification of the Bayesian Panel VAR model with country grouping for global prior parameters BVARGROUPPRIORPANEL. The groups can be pre-specified, which requires the argument `group_allocation` to be provided, or estimated, which requires the argument `G` for the number of groups to be provided and the argument `group_allocation` to be left empty.

Usage:

```
BVARGROUPPRIORPANEL$new(
  data,
  p = 1L,
  exogenous = NULL,
  stationary = rep(FALSE, ncol(data[[1]])),
  type = rep("real", ncol(data[[1]])),
  G = NULL,
  group_allocation = NULL
)
```

Arguments:

`data` a list with `C` elements of `(T_c+p)xN` matrices with time series data.

`p` a positive integer providing model's autoregressive lag order.

`exogenous` a `(T+p)xd` matrix of exogenous variables.

`stationary` an `N` logical vector - its element set to `FALSE` sets the prior mean for the autoregressive parameters of the `Nth` equation to the white noise process, otherwise to random walk.

`type` an `N` character vector with elements set to "rate" or "real" determining the truncation of the predictive density to `[0, 100]` and `(-Inf, Inf)` (no truncation) for each of the variables.

`G` a positive integer specifying the number of country groups. Its specification is required if `group_allocation` is not provided and the country groups to be estimated.

`group_allocation` an argument that can be provided as a numeric vector with integer numbers denoting group allocations to pre-specify the the country groups, in which case they are not estimated, or left empty if the country groups are to be estimated.

Returns: A new complete specification for the Bayesian Panel VAR model BVARPANEL.

`BVARGROUPPRIORPANEL$get_data_matrices()`: Returns the data matrices as the DataMatricesBVARPANEL object.

Usage:

```
BVARGROUPPRIORPANEL$get_data_matrices()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_data_matrices()
```

`BVARGROUPPRIORPANEL$get_prior()`: Returns the prior specification as the PriorBVARPANEL object.

Usage:

```
BVARGROUPPRIORPANEL$get_prior()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_prior()
```

`BVARGROUPPRIORPANEL$get_starting_values()`: Returns the starting values as the StartingValuesBVARPANEL object.

Usage:

```
BVARGROUPPRIORPANEL$get_starting_values()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_starting_values()
```

`BVARGROUPPRIORPANEL$get_type()`: Returns the type of the model.

Usage:

```
BVARGROUPPRIORPANEL$get_type()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_type()
```

`BVARGROUPPRIORPANEL$set_global2pooled()`: Sets the prior mean of the global autoregressive parameters to the OLS pooled panel estimator following Zellner, Hong (1989).

Usage:

```
BVARGROUPPRIORPANEL$set_global2pooled(x)
```

Arguments:

`x` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$set_global2pooled()
```

`BVARGROUPPRIORPANEL$set_adaptiveMH()`**:** Sets the parameters of adaptive Metropolis-Hastings sampler for the parameter nu.

Usage:

```
BVARGROUPPRIORPANEL$set_adaptiveMH(x)
```

Arguments:

`x` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel[1:5]
)
spec$set_adaptiveMH(c(0.6, 0.4, 10, 0.1))
```

`BVARGROUPPRIORPANEL$clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
BVARGROUPPRIORPANEL$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

References

Zellner, Hong (1989). Forecasting international growth rates using Bayesian shrinkage and other procedures. *Journal of Econometrics*, **40**(1), 183–202, doi:10.1016/03044076(89)900365.

Examples

```
spec = specify_bvarGroupPriorPANEL$new(
   data = ilo_dynamic_panel,
   G = 2
)


## ------------------------------------------------
## Method `BVARGROUPPRIORPANEL$get_data_matrices()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_data_matrices()
```

```
## ------------------------------------------------
## Method `BVARGROUPPRIORPANEL$get_prior()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_prior()


## ------------------------------------------------
## Method `BVARGROUPPRIORPANEL$get_starting_values()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_starting_values()


## ------------------------------------------------
## Method `BVARGROUPPRIORPANEL$get_type()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_type()


## ------------------------------------------------
## Method `BVARGROUPPRIORPANEL$set_global2pooled()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$set_global2pooled()


## ------------------------------------------------
## Method `BVARGROUPPRIORPANEL$set_adaptiveMH()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel[1:5]
)
spec$set_adaptiveMH(c(0.6, 0.4, 10, 0.1))
```

`specify_bvarPANEL` *R6 Class representing the specification of the BVARPANEL model*

Description

The class BVARPANEL presents complete specification for the Bayesian Panel Vector Autoregression.

Public fields

`p` a non-negative integer specifying the autoregressive lag order of the model.

`prior` an object PriorBSVAR with the prior specification.

`data_matrices` an object DataMatricesBVARPANEL with the data matrices.

`starting_values` an object StartingValuesBVARPANEL with the starting values.

`adaptiveMH` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Methods

Public methods:

- `BVARPANEL$new()`
- `BVARPANEL$set_to_Jarocinski()`
- `BVARPANEL$get_data_matrices()`
- `BVARPANEL$get_prior()`
- `BVARPANEL$get_starting_values()`
- `BVARPANEL$get_type()`
- `BVARPANEL$set_global2pooled()`
- `BVARPANEL$set_adaptiveMH()`
- `BVARPANEL$clone()`

`BVARPANEL$new()`: Create a new specification of the Bayesian Panel VAR model BVARPANEL.

Usage:

```
BVARPANEL$new(
  data,
  p = 1L,
  exogenous = NULL,
  stationary = rep(FALSE, ncol(data[[1]])),
  type = rep("real", ncol(data[[1]]))
)
```

Arguments:

`data` a list with `C` elements of `(T_c+p)xN` matrices with time series data.

`p` a positive integer providing model's autoregressive lag order.

`exogenous` a `(T+p)xd` matrix of exogenous variables.

`stationary` an `N` logical vector - its element set to `FALSE` sets the prior mean for the autoregressive parameters of the `N`th equation to the white noise process, otherwise to random walk.

`type` an `N` character vector with elements set to "rate" or "real" determining the truncation of the predictive density to `[0, 100]` and `(-Inf, Inf)` (no truncation) for each of the variables.

Returns: A new complete specification for the Bayesian Panel VAR model BVARPANEL.

`BVARPANEL$set_to_Jarocinski()`: Sets the model in line with the specification by Jarocinski (2010) as presented by Dieppe, Legrand, Roye (2016).

Usage:

```
BVARPANEL$set_to_Jarocinski()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$set_to_Jarocinski()
```

`BVARPANEL$get_data_matrices()`: Returns the data matrices as the DataMatricesBVARPANEL object.

Usage:

```
BVARPANEL$get_data_matrices()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_data_matrices()
```

`BVARPANEL$get_prior()`: Returns the prior specification as the PriorBVARPANEL object.

Usage:

```
BVARPANEL$get_prior()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_prior()
```

`BVARPANEL$get_starting_values()`: Returns the starting values as the StartingValuesBVARPANEL object.

Usage:

```
BVARPANEL$get_starting_values()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_starting_values()
```

`BVARPANEL$get_type()`**:** Returns the type of the model.

Usage:

```
BVARPANEL$get_type()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_type()
```

`BVARPANEL$set_global2pooled()`**:** Sets the prior mean of the global autoregressive parameters to the OLS pooled panel estimator following Zellner, Hong (1989).

Usage:

```
BVARPANEL$set_global2pooled(x)
```

Arguments:

`x` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$set_global2pooled()
```

`BVARPANEL$set_adaptiveMH()`**:** Sets the parameters of adaptive Metropolis-Hastings sampler for the parameter nu.

Usage:

```
BVARPANEL$set_adaptiveMH(x)
```

Arguments:

`x` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$set_adaptiveMH(c(0.6, 0.4, 10, 0.1))
```

`BVARPANEL$clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
BVARPANEL$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

References

Jarocinski (2010). Responses to monetary policy shocks in the east and the west of Europe: a comparison. *Journal of Applied Econometrics*, **25**(5), 833-868, doi:10.1002/jae.1082.

Dieppe, Legrand, Roye (2016). The BEAR toolbox, *ECB Working Papers*, **1934**, doi:10.2866/292952.

Zellner, Hong (1989). Forecasting international growth rates using Bayesian shrinkage and other procedures. *Journal of Econometrics*, **40**(1), 183–202, doi:10.1016/03044076(89)900365.

Examples

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)


## ------------------------------------------------
## Method `BVARPANEL$set_to_Jarocinski()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$set_to_Jarocinski()


## ------------------------------------------------
## Method `BVARPANEL$get_data_matrices()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_data_matrices()


## ------------------------------------------------
## Method `BVARPANEL$get_prior()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_prior()


## ------------------------------------------------
## Method `BVARPANEL$get_starting_values()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
```

```
)
spec$get_starting_values()


## ------------------------------------------------
## Method `BVARPANEL$get_type()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_type()


## ------------------------------------------------
## Method `BVARPANEL$set_global2pooled()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$set_global2pooled()


## ------------------------------------------------
## Method `BVARPANEL$set_adaptiveMH()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$set_adaptiveMH(c(0.6, 0.4, 10, 0.1))
```

`specify_bvars` *R6 Class representing the specification of the BVARs model*

Description

The class BVARs presents complete specification for the Bayesian Vector Autoregressions for cubic data.

Public fields

`p` a non-negative integer specifying the autoregressive lag order of the model.

`prior` an object PriorBSVAR with the prior specification.

`data_matrices` an object DataMatricesBVARPANEL with the data matrices.

`starting_values` an object StartingValuesBVARPANEL with the starting values.

`adaptiveMH` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Methods

Public methods:

- BVARs$new()
- BVARs$get_data_matrices()
- BVARs$get_prior()
- BVARs$get_starting_values()
- BVARs$get_type()
- BVARs$set_prior2objective()
- BVARs$set_global2pooled()
- BVARs$set_adaptiveMH()
- BVARs$clone()

BVARs$new(): Create a new specification of the Bayesian Panel VAR model BVARPANEL.

Usage:

```
BVARs$new(
  data,
  p = 1L,
  exogenous = NULL,
  stationary = rep(FALSE, ncol(data[[1]])),
  type = rep("real", ncol(data[[1]]))
)
```

Arguments:

`data` a list with `C` elements of `(T_c+p)xN` matrices with time series data.

`p` a positive integer providing model's autoregressive lag order.

`exogenous` a `(T+p)xd` matrix of exogenous variables.

`stationary` an `N` logical vector - its element set to `FALSE` sets the prior mean for the autoregressive parameters of the `N`th equation to the white noise process, otherwise to random walk.

`type` an `N` character vector with elements set to "rate" or "real" determining the truncation of the predictive density to `[0, 100]` and `(-Inf, Inf)` (no truncation) for each of the variables.

Returns: A new complete specification for the Bayesian Panel VAR model BVARPANEL.

BVARs$get_data_matrices(): Returns the data matrices as the DataMatricesBVARPANEL object.

Usage:

```
BVARs$get_data_matrices()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_data_matrices()
```

BVARs$get_prior(): Returns the prior specification as the PriorBVARPANEL object.

Usage:

```
BVARs$get_prior()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_prior()
```

`BVARs$get_starting_values()`: Returns the starting values as the StartingValuesBVARPANEL object.

Usage:

```
BVARs$get_starting_values()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_starting_values()
```

`BVARs$get_type()`: Returns the type of the model.

Usage:

```
BVARs$get_type()
```

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_type()
```

`BVARs$set_prior2objective()`: Sets the VAR model priors to objective prior by Zellner (1972).

Usage:

```
BVARs$set_prior2objective()
```

Examples:

```
spec = specify_bvars$new(
   data = ilo_dynamic_panel
)
spec$set_prior2objective()
```

`BVARs$set_global2pooled()`: Sets the prior mean of the global autoregressive parameters to the OLS pooled panel estimator following Zellner, Hong (1989).

Usage:

```
BVARs$set_global2pooled(x)
```

Arguments:

`x` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$set_global2pooled()
```

`BVARs$set_adaptiveMH()`: Sets the parameters of adaptive Metropolis-Hastings sampler for the parameter nu.

Usage:

```
BVARs$set_adaptiveMH(x)
```

Arguments:

`x` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Examples:

```
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$set_adaptiveMH(c(0.6, 0.4, 10, 0.1))
```

`BVARs$clone()`: The objects of this class are cloneable with this method.

Usage:

```
BVARs$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

References

Zellner (1971). *An Introduction to Bayesian Inference in Econometrics*. John Wiley & Sons.

Zellner, Hong (1989). Forecasting international growth rates using Bayesian shrinkage and other procedures. *Journal of Econometrics*, **40**(1), 183–202, doi:10.1016/03044076(89)900365.

Examples

```
spec = specify_bvars$new(
   data = ilo_dynamic_panel
)


## ------------------------------------------------
## Method `BVARs$get_data_matrices()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_data_matrices()
```

```
## ------------------------------------------------
## Method `BVARs$get_prior()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_prior()


## ------------------------------------------------
## Method `BVARs$get_starting_values()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_starting_values()


## ------------------------------------------------
## Method `BVARs$get_type()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$get_type()


## ------------------------------------------------
## Method `BVARs$set_prior2objective()`
## ------------------------------------------------

spec = specify_bvars$new(
   data = ilo_dynamic_panel
)
spec$set_prior2objective()


## ------------------------------------------------
## Method `BVARs$set_global2pooled()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$set_global2pooled()


## ------------------------------------------------
## Method `BVARs$set_adaptiveMH()`
```

```
## ------------------------------------------------

spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel
)
spec$set_adaptiveMH(c(0.6, 0.4, 10, 0.1))
```

---

specify_panel_data_matrices

R6 Class Representing DataMatricesBVARPANEL

Description

The class DataMatricesBVARPANEL presents the data matrices of dependent variables, $\mathbf{Y}_c$, and regressors, $\mathbf{X}_c$, for the Bayesian Panel VAR model for all countries $c = 1, ..., C$.

Public fields

`Y` a list with `C` elements with `(T_c + p) x N` matrices of dependent variables, $\mathbf{Y}_c$, possibly with missing observations given by `NA`.

`missing` a list with `C` elements with `T_c x N` matrices containing value `1` for missing observation and `0` otherwise.

`type` an `N` character vector with elements set to "rate" or "real" determining the truncation of the predictive density to `[0, 100]` and `(-Inf, Inf)` (no truncation) for each of the variables.

`exogenous` a list with `C` elements with `(T_c + p) x N` matrices of exogenous variables.

Methods

Public methods:

- `DataMatricesBVARPANEL$new()`
- `DataMatricesBVARPANEL$get_data_matrices()`
- `DataMatricesBVARPANEL$clone()`

`DataMatricesBVARPANEL$new()`: Create new data matrices DataMatricesBVARPANEL

Usage:

```
DataMatricesBVARPANEL$new(
  data,
  p = 1L,
  exogenous = NULL,
  type = rep("real", ncol(data[[1]]))
)
```

Arguments:

`data` a list containing `(T_c+p)xN` matrices with country-specific time series data.

`p` a positive integer providing model's autoregressive lag order.

`exogenous` a list containing `(T_c+p)xd` matrices with country-specific of exogenous variables. This matrix should not include a constant term.

`type` an `N` character vector with elements set to "rate" or "real" determining the truncation of the predictive density to `[0, 100]` and `(-Inf, Inf)` (no truncation) for each of the variables.

Returns: New data matrices DataMatricesBVARPANEL

`DataMatricesBVARPANEL$get_data_matrices()`**:** Returns the data matrices DataMatricesBVARPANEL as a `list`.

Usage:

```
DataMatricesBVARPANEL$get_data_matrices()
```

Examples:

```
data(ilo_dynamic_panel)
YX = specify_panel_data_matrices$new(ilo_dynamic_panel)
YX$get_data_matrices()
```

`DataMatricesBVARPANEL$clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
DataMatricesBVARPANEL$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
data(ilo_dynamic_panel)
YX = specify_panel_data_matrices$new(data = ilo_dynamic_panel, p = 4)
length(YX$Y); names(YX$Y)


## ------------------------------------------------
## Method `DataMatricesBVARPANEL$get_data_matrices()`
## ------------------------------------------------

data(ilo_dynamic_panel)
YX = specify_panel_data_matrices$new(ilo_dynamic_panel)
YX$get_data_matrices()
```

`specify_poosf_exercise` *R6 Class Representing specification of the pseudo-out-of-sample forecasting exercise*

Description

The class `POOSForecastSetup` presents the specification of the recursive pseudo-out-of-sample forecasting exercise using expanding window samples.

Public fields

`S` a positive integer number - the number of MCMC draws from the posterior distribution in the estimation for each forecasts

`S_burn` a positive integer number - the number of MCMC draws from to achieve convergence in the estimation for each forecasts

`horizons` a vector with positive integer numbers - the forecast horizons used in the forecast performance evaluation

`training_sample` a positive integer number - the number of of the first observations to be used in the estimation for the first forecast

Methods

Public methods:

- `POOSForecastSetup$new()`
- `POOSForecastSetup$clone()`

`POOSForecastSetup$new()`: Create a new specification `POOSForecastSetup` of the recursive pseudo-out-of-sample forecasting exercise using expanding window samples.

Usage:

```
POOSForecastSetup$new(spec, S, S_burn, horizons = 1L, training_sample = 1L)
```

Arguments:

`spec` a model specification object of class `BVARPANEL` or `BVARGROUPPANEL`.

`S` a positive integer - number of draws from the posterior distribution for model estimation for each of the forecasts

`S_burn` a positive integer - number of draws from the posterior distribution to obtain convergence for model estimation for each of the forecasts

`horizons` a vector with positive integers - forecast horizons at which forecasting performance is to be verified

`training_sample` a positive integer - the number of the first observations to be used for estimation for the first forecast. The number of observations to be used for remaining forecasts increases recursively by one for each forecast.

Returns: A new specification `POOSForecastSetup` of the recursive pseudo-out-of-sample forecasting exercise using expanding window samples.

Examples:

```
spec = specify_bvarPANEL$new(ilo_dynamic_panel)
poos = specify_poosf_exercise$new(spec, 5000, 1000, 1:2, 10)
```

`POOSForecastSetup$clone()`: The objects of this class are cloneable with this method.

Usage:

```
POOSForecastSetup$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
spec = specify_bvarPANEL$new(ilo_dynamic_panel)
poos = specify_poosf_exercise$new(spec, 5000, 1000, 1:2, 10)


## ------------------------------------------------
## Method `POOSForecastSetup$new()`
## ------------------------------------------------

spec = specify_bvarPANEL$new(ilo_dynamic_panel)
poos = specify_poosf_exercise$new(spec, 5000, 1000, 1:2, 10)
```

`specify_posterior_bvarGroupPANEL`
R6 Class Representing PosteriorBVARGROUPPANEL

Description

The class PosteriorBVARGROUPPANEL contains posterior output and the specification including the last MCMC draw for the Bayesian Panel VAR model with country grouping. Note that due to the thinning of the MCMC output the starting value in element `last_draw` might not be equal to the last draw provided in element `posterior`.

Super class

`PosteriorBVARPANEL` -> `PosteriorBVARGROUPPANEL`

Public fields

`last_draw` an object of class BVARGROUPPANEL with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using `estimate()`.

`posterior` a list containing Bayesian estimation output.

Methods

Public methods:

- `PosteriorBVARGROUPPANEL$new()`
- `PosteriorBVARGROUPPANEL$clone()`

`PosteriorBVARGROUPPANEL$new()`: Create a new posterior output PosteriorBVARGROUPPANEL.

Usage:

`PosteriorBVARGROUPPANEL$new(specification, posterior)`

Arguments:

`specification` an object of class BVARGROUPPANEL with the last draw of the current MCMC run as the starting value.

`posterior` a list containing Bayesian estimation output.

Returns: A posterior output PosteriorBVARGROUPPANEL.

`PosteriorBVARGROUPPANEL$clone()`: The objects of this class are cloneable with this method.

Usage:

`PosteriorBVARGROUPPANEL$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

See Also

specify_bvarGroupPANEL

Examples

```
spec = specify_bvarGroupPANEL$new(
   data = ilo_dynamic_panel[1:5],
   G = 2
)
#posterior       = estimate(specification, 5)
#class(posterior)
```

`specify_posterior_bvarGroupPriorPANEL`
R6 Class Representing PosteriorBVARGROUPPRIORPANEL

Description

The class PosteriorBVARGROUPPRIORPANEL contains posterior output and the specification including the last MCMC draw for the Bayesian Panel VAR model with country grouping. Note that due to the thinning of the MCMC output the starting value in element `last_draw` might not be equal to the last draw provided in element `posterior`.

Super classes

`PosteriorBVARPANEL` -> `PosteriorBVARGROUPPANEL` -> `PosteriorBVARGROUPPRIORPANEL`

Public fields

`last_draw` an object of class BVARGROUPPRIORPANEL with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using `estimate()`.

`posterior` a list containing Bayesian estimation output.

Methods

Public methods:

- PosteriorBVARGROUPPRIORPANEL$new()
- PosteriorBVARGROUPPRIORPANEL$clone()

PosteriorBVARGROUPPRIORPANEL$new(): Create a new posterior output PosteriorBVARGROUP-PRIORPANEL.

Usage:

```
PosteriorBVARGROUPPRIORPANEL$new(specification, posterior)
```

Arguments:

specification an object of class BVARGROUPPRIORPANEL with the last draw of the current MCMC run as the starting value.

posterior a list containing Bayesian estimation output.

Returns: A posterior output PosteriorBVARGROUPPRIORPANEL.

PosteriorBVARGROUPPRIORPANEL$clone(): The objects of this class are cloneable with this method.

Usage:

```
PosteriorBVARGROUPPRIORPANEL$clone(deep = FALSE)
```

Arguments:

deep Whether to make a deep clone.

See Also

specify_bvarGroupPriorPANEL

Examples

```
spec = specify_bvarGroupPriorPANEL$new(
   data = ilo_dynamic_panel[1:5],
   G = 2
)
#posterior       = estimate(specification, 5)
#class(posterior)
```

---

specify_posterior_bvarPANEL

R6 Class Representing PosteriorBVARPANEL

Description

The class PosteriorBVARPANEL contains posterior output and the specification including the last MCMC draw for the Bayesian Panel VAR model. Note that due to the thinning of the MCMC output the starting value in element `last_draw` might not be equal to the last draw provided in element `posterior`.

Public fields

`last_draw` an object of class BVARPANEL with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using `estimate()`.

`posterior` a list containing Bayesian estimation output.

Methods

Public methods:

- `PosteriorBVARPANEL$new()`
- `PosteriorBVARPANEL$get_posterior()`
- `PosteriorBVARPANEL$get_last_draw()`
- `PosteriorBVARPANEL$clone()`

`PosteriorBVARPANEL$new()`: Create a new posterior output PosteriorBVARPANEL.

Usage:

```
PosteriorBVARPANEL$new(specification_bvarPANEL, posterior_bvarPANEL)
```

Arguments:

`specification_bvarPANEL` an object of class BVARPANEL with the last draw of the current MCMC run as the starting value.

`posterior_bvarPANEL` a list containing Bayesian estimation output.

Returns: A posterior output PosteriorBVARPANEL.

`PosteriorBVARPANEL$get_posterior()`: Returns a list containing Bayesian estimation output.

Usage:

```
PosteriorBVARPANEL$get_posterior()
```

Examples:

```
specification = specify_bvarPANEL$new(
   data = ilo_dynamic_panel[1:5]
)
posterior       = estimate(specification, 5)
posterior$get_posterior()
```

`PosteriorBVARPANEL$get_last_draw()`: Returns an object of class BVARPANEL with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using `estimate()`.

Usage:

```
PosteriorBVARPANEL$get_last_draw()
```

Examples:

```
specification = specify_bvarPANEL$new(
   data = ilo_dynamic_panel[1:5]
)
burn_in        = estimate(specification, 5)
posterior      = estimate(burn_in, 5)
```

`PosteriorBVARPANEL$clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
PosteriorBVARPANEL$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

See Also

specify_bvarPANEL

Examples

```
specification = specify_bvarPANEL$new(
   data = ilo_dynamic_panel[1:5]
)
posterior       = estimate(specification, 5)
class(posterior)


## ------------------------------------------------
## Method `PosteriorBVARPANEL$get_posterior()`
## ------------------------------------------------

specification = specify_bvarPANEL$new(
   data = ilo_dynamic_panel[1:5]
)
posterior       = estimate(specification, 5)
posterior$get_posterior()


## ------------------------------------------------
## Method `PosteriorBVARPANEL$get_last_draw()`
## ------------------------------------------------

specification = specify_bvarPANEL$new(
   data = ilo_dynamic_panel[1:5]
)
burn_in        = estimate(specification, 5)
posterior      = estimate(burn_in, 5)
```

`specify_posterior_bvars` *R6 Class Representing PosteriorBVARs*

Description

The class PosteriorBVARs contains posterior output and the specification including the last MCMC draw for the Bayesian Panel VAR model. Note that due to the thinning of the MCMC output the starting value in element `last_draw` might not be equal to the last draw provided in element `posterior`.

Public fields

`last_draw` an object of class BVARs with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using `estimate()`.

`posterior` a list containing Bayesian estimation output.

Methods

Public methods:

- `PosteriorBVARs$new()`
- `PosteriorBVARs$get_posterior()`
- `PosteriorBVARs$get_last_draw()`
- `PosteriorBVARs$clone()`

`PosteriorBVARs$new()`: Create a new posterior output PosteriorBVARs.

Usage:

```
PosteriorBVARs$new(specification_bvarPANEL, posterior_bvarPANEL)
```

Arguments:

`specification_bvarPANEL` an object of class BVARs with the last draw of the current MCMC run as the starting value.

`posterior_bvarPANEL` a list containing Bayesian estimation output.

Returns: A posterior output PosteriorBVARs.

`PosteriorBVARs$get_posterior()`: Returns a list containing Bayesian estimation output.

Usage:

```
PosteriorBVARs$get_posterior()
```

Examples:

```
specification = specify_bvars$new(
   data = ilo_dynamic_panel[1:5]
)
posterior       = estimate(specification, 5)
posterior$get_posterior()
```

`PosteriorBVARs$get_last_draw()`: Returns an object of class BVARs with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using `estimate()`.

Usage:

```
PosteriorBVARs$get_last_draw()
```

Examples:

```
specification = specify_bvars$new(
   data = ilo_dynamic_panel[1:5]
)
burn_in        = estimate(specification, 5)
posterior      = estimate(burn_in, 5)
```

`PosteriorBVARs$clone()`: The objects of this class are cloneable with this method.

Usage:

```
PosteriorBVARs$clone(deep = FALSE)
```

Arguments:

deep Whether to make a deep clone.

See Also

specify_bvars

Examples

```
specification = specify_bvars$new(
   data = ilo_dynamic_panel[1:5]
)
posterior       = estimate(specification, 5)
class(posterior)


## ------------------------------------------------
## Method `PosteriorBVARs$get_posterior()`
## ------------------------------------------------

specification = specify_bvars$new(
   data = ilo_dynamic_panel[1:5]
)
posterior       = estimate(specification, 5)
posterior$get_posterior()


## ------------------------------------------------
## Method `PosteriorBVARs$get_last_draw()`
## ------------------------------------------------

specification = specify_bvars$new(
   data = ilo_dynamic_panel[1:5]
)
burn_in        = estimate(specification, 5)
posterior      = estimate(burn_in, 5)
```

specify_prior_bvarPANEL

R6 Class Representing PriorBVARPANEL

Description

The class PriorBVARPANEL presents a prior specification for the Bayesian hierarchical panel VAR model.

Public fields

`M` an `KxN` matrix, the mean of the second-level MNIW prior distribution for the global parameter matrices $\mathbf{A}$ and $\mathbf{V}$

`W` a `KxK` column-specific covariance matrix of the second-level MNIW prior distribution for the global parameter matrices $\mathbf{A}$ and $\mathbf{V}$

`S_inv` an `NxN` row-specific precision matrix of the second-level MNIW prior distribution for the global parameter matrices $\mathbf{A}$ and $\mathbf{V}$

`S_Sigma_inv` an `NxN` precision matrix of the second-level Wishart prior distribution for the global parameter matrix $\mathbf{\Sigma}$.

`eta` a positive shape parameter of the second-level MNIW prior distribution for the global parameter matrices $\mathbf{A}$ and $\mathbf{V}$

`mu_Sigma` a positive shape parameter of the second-level Wishart prior distribution for the global parameter matrix $\mathbf{\Sigma}$.

`lambda` a positive shape of the second-level exp prior distribution for the shape parameter ν.

`mu_m` a scalar mean of the third-level normal prior distribution for the global average persistence parameter m.

`sigma2_m` a positive scalar variance of the third-level normal prior distribution for the global average persistence parameter m.

`s_w` a positive scalar scale of the third-level gamma prior distribution for parameter w.

`a_w` a positive scalar shape of the third-level gamma prior distribution for parameter w.

`s_s` a positive scalar scale parameter of the third-level inverted-gamma 2 prior distribution for parameter s.

`nu_s` a positive scalar shape parameter of the third-level inverted-gamma 2 prior distribution for parameter s.

Methods

Public methods:

- `PriorBVARPANEL$new()`
- `PriorBVARPANEL$get_prior()`
- `PriorBVARPANEL$clone()`

`PriorBVARPANEL$new():` Create a new prior specification PriorBVARPANEL.

Usage:

```
PriorBVARPANEL$new(C, N, p, d = 0, stationary = rep(FALSE, N))
```

Arguments:

`C` a positive integer - the number of countries in the data.

`N` a positive integer - the number of dependent variables in the model.

`p` a positive integer - the autoregressive lag order of the SVAR model.

`d` a positive integer - the number of `exogenous` variables in the model.

`stationary` an `N` logical vector - its element set to `FALSE` sets the prior mean for the autoregressive parameters of the `N`th equation to the white noise process, otherwise to random walk.

Returns: A new prior specification PriorBVARPANEL.

Examples:

```
# a prior for 2-country, 3-variable example with one lag and stationary data
prior = specify_prior_bvarPANEL$new(C = 2, N = 3, p = 1)
prior$M
```

`PriorBVARPANEL$get_prior()`: Returns the elements of the prior specification PriorBSVAR as a `list`.

Usage:

```
PriorBVARPANEL$get_prior()
```

Examples:

```
# a prior for 2-coutnry, 3-variable example with four lags
prior = specify_prior_bvarPANEL$new(C = 2, N = 3, p = 4)
prior$get_prior() # show the prior as list
```

`PriorBVARPANEL$clone()`: The objects of this class are cloneable with this method.

Usage:

```
PriorBVARPANEL$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
prior = specify_prior_bvarPANEL$new(C = 2, N = 3, p = 1)
prior$M


## ------------------------------------------------
## Method `PriorBVARPANEL$new()`
## ------------------------------------------------

# a prior for 2-country, 3-variable example with one lag and stationary data
prior = specify_prior_bvarPANEL$new(C = 2, N = 3, p = 1)
prior$M
```

```
## ------------------------------------------------
## Method `PriorBVARPANEL$get_prior()`
## ------------------------------------------------

# a prior for 2-coutnry, 3-variable example with four lags
prior = specify_prior_bvarPANEL$new(C = 2, N = 3, p = 4)
prior$get_prior() # show the prior as list
```

---

specify_prior_bvars *R6 Class Representing PriorBVARs*

Description

The class PriorBVARs presents a prior specification for the Bayesian VAR model for each country.

Public fields

M an KxN matrix, the mean of the MNIW prior distribution for the autoregressive matrices $\mathbf{A}_c$

W a KxK column-specific covariance matrix of the MNIW prior distribution for the autoregressive matrices $\mathbf{A}_c$

S_inv an NxN row-specific precision matrix of the MNIW prior distribution for the covariance matrices $\mathbf{\Sigma}_c$

lambda a positive shape of the exponential prior distribution for the shape parameter ν.

mu_m a scalar mean of the normal prior distribution for the average persistence parameter m.

sigma2_m a positive scalar variance of the normal prior distribution for the average persistence parameter m.

s_w a positive scalar scale of the inverse-gamma 2 prior distribution for parameter w.

nu_w a positive scalar shape of the inverse-gamma 2 prior distribution for parameter w.

s_s a positive scalar scale parameter of the gamma prior distribution for parameter s.

a_s a positive scalar shape parameter of the gamma prior distribution for parameter s.

Methods

Public methods:

- PriorBVARs$new()
- PriorBVARs$get_prior()
- PriorBVARs$clone()

PriorBVARs$new(): Create a new prior specification PriorBVARs.

Usage:

```
PriorBVARs$new(C, N, p, d = 0, stationary = rep(FALSE, N))
```

Arguments:

`C` a positive integer - the number of countries in the data.

`N` a positive integer - the number of dependent variables in the model.

`p` a positive integer - the autoregressive lag order of the SVAR model.

`d` a positive integer - the number of `exogenous` variables in the model.

`stationary` an `N` logical vector - its element set to `FALSE` sets the prior mean for the autoregressive parameters of the `N`th equation to the white noise process, otherwise to random walk.

Returns: A new prior specification PriorBVARs.

Examples:

```
# a prior for 2-country, 3-variable example with one lag and stationary data
prior = specify_prior_bvars$new(C = 2, N = 3, p = 1)
prior$M
```

`PriorBVARs$get_prior()`: Returns the elements of the prior specification PriorBVARs as a `list`.

Usage:

```
PriorBVARs$get_prior()
```

Examples:

```
# a prior for 2-country, 3-variable example with four lags
prior = specify_prior_bvars$new(C = 2, N = 3, p = 4)
prior$get_prior() # show the prior as list
```

`PriorBVARs$clone()`: The objects of this class are cloneable with this method.

Usage:

```
PriorBVARs$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
prior = specify_prior_bvars$new(C = 2, N = 3, p = 1)
prior$M


## ------------------------------------------------
## Method `PriorBVARs$new()`
## ------------------------------------------------

# a prior for 2-country, 3-variable example with one lag and stationary data
prior = specify_prior_bvars$new(C = 2, N = 3, p = 1)
prior$M


## ------------------------------------------------
## Method `PriorBVARs$get_prior()`
## ------------------------------------------------
```

```
# a prior for 2-country, 3-variable example with four lags
prior = specify_prior_bvars$new(C = 2, N = 3, p = 4)
prior$get_prior() # show the prior as list
```

---

specify_starting_values_bvarGroupPANEL
R6 Class Representing StartingValuesBVARGROUPPANEL

Description

The class StartingValuesBVARGROUPPANEL presents starting values for the Bayesian hierarchical panel VAR model with country grouping

Super class

`StartingValuesBVARPANEL` -> `StartingValuesBVARGROUPPANEL`

Public fields

`group_allocation` a numeric vector with integer numbers denoting group allocations

`A_c` an `KxNxC` array of starting values for the local parameter $\mathbf{A}_c$.

`Sigma_c` an `NxNxC` array of starting values for the local parameter $\mathbf{\Sigma}_c$.

`A_g` an `KxNxG` array of starting values for the group parameter $\mathbf{A}_g$.

`Sigma_g` an `NxNxG` array of starting values for the group parameter $\mathbf{\Sigma}_g$.

`A` an `KxN` matrix of starting values for the global parameter $\mathbf{A}$.

`V` an `KxK` matrix of starting values for the global parameter $\mathbf{V}$.

`Sigma` an `NxN` matrix of starting values for the global parameter $\mathbf{\Sigma}$.

`nu` a positive scalar with starting values for the global parameter ν.

`m` a positive scalar with starting values for the global hyper-parameter m.

`w` a positive scalar with starting values for the global hyper-parameter w.

`s` a positive scalar with starting values for the global hyper-parameter s.

Methods

Public methods:

- `StartingValuesBVARGROUPPANEL$new()`
- `StartingValuesBVARGROUPPANEL$get_starting_values()`
- `StartingValuesBVARGROUPPANEL$set_starting_values()`
- `StartingValuesBVARGROUPPANEL$clone()`

`StartingValuesBVARGROUPPANEL$new()`: Create new starting values StartingValuesBVARGROUPPANEL

Usage:

```
StartingValuesBVARGROUPPANEL$new(group_allocation = 1:C, C, G = C, N, p, d = 0)
```

Arguments:

`group_allocation` a numeric vector with integer numbers denoting group allocations

`C` a positive integer - the number of countries in the data.

`G` a positive integer specifying the number of country groups.

`N` a positive integer - the number of dependent variables in the model.

`p` a positive integer - the autoregressive lag order of the SVAR model.

`d` a positive integer - the number of `exogenous` variables in the model.

Returns: Starting values StartingValuesBVARGROUPPANEL

Examples:

```
# starting values for Bayesian Panel VAR 2-country model with 4 lags for a 3-variable system.
sv = specify_starting_values_bvarGroupPANEL$new(C = 2, N = 3, p = 1)
```

`StartingValuesBVARGROUPPANEL$get_starting_values()`: Returns the elements of the starting values StartingValuesBVARGROUPPANEL as a `list`.

Usage:

```
StartingValuesBVARGROUPPANEL$get_starting_values()
```

Examples:

```
# starting values for a homoskedastic bsvar with 1 lag for a 3-variable system
sv = specify_starting_values_bvarGroupPANEL$new(rep(1,2), C = 2, N = 3, p = 1)
sv$get_starting_values()   # show starting values as list
```

`StartingValuesBVARGROUPPANEL$set_starting_values()`: Returns the elements of the starting values StartingValuesBVARGROUPPANEL as a `list`.

Usage:

```
StartingValuesBVARGROUPPANEL$set_starting_values(last_draw)
```

Arguments:

`last_draw` a list containing the same elements as object StartingValuesBVARGROUPPANEL

Returns: An object of class StartingValuesBVARGROUPPANEL including the last draw of the current MCMC as the starting value to be passed to the continuation of the MCMC estimation.

Examples:

```
sv = specify_starting_values_bvarGroupPANEL$new(rep(1,2), C = 2, N = 3, p = 1)

# Modify the starting values by:
sv_list = sv$get_starting_values()   # getting them as list
sv_list$A <- matrix(rnorm(12), 3, 4) # modifying the entry
sv$set_starting_values(sv_list)      # providing to the class object
```

`StartingValuesBVARGROUPPANEL$clone()`: The objects of this class are cloneable with this method.

Usage:

```
StartingValuesBVARGROUPPANEL$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
# starting values for a Bayesian Panel VAR
sv = specify_starting_values_bvarGroupPANEL$new(rep(1,2), C = 2, G = 1, N = 3, p = 1)


## ------------------------------------------------
## Method `StartingValuesBVARGROUPPANEL$new()`
## ------------------------------------------------

# starting values for Bayesian Panel VAR 2-country model with 4 lags for a 3-variable system.
sv = specify_starting_values_bvarGroupPANEL$new(C = 2, N = 3, p = 1)


## ------------------------------------------------
## Method `StartingValuesBVARGROUPPANEL$get_starting_values()`
## ------------------------------------------------

# starting values for a homoskedastic bsvar with 1 lag for a 3-variable system
sv = specify_starting_values_bvarGroupPANEL$new(rep(1,2), C = 2, N = 3, p = 1)
sv$get_starting_values()   # show starting values as list


## ------------------------------------------------
## Method `StartingValuesBVARGROUPPANEL$set_starting_values()`
## ------------------------------------------------

sv = specify_starting_values_bvarGroupPANEL$new(rep(1,2), C = 2, N = 3, p = 1)

# Modify the starting values by:
sv_list = sv$get_starting_values()   # getting them as list
sv_list$A <- matrix(rnorm(12), 3, 4) # modifying the entry
sv$set_starting_values(sv_list)      # providing to the class object
```

---

specify_starting_values_bvarGroupPriorPANEL
R6 Class Representing StartingValuesBVARGROUPPRIORPANEL

Description

The class StartingValuesBVARGROUPPRIORPANEL presents starting values for the Bayesian hierarchical panel VAR model with country grouping

Public fields

`group_allocation` a numeric vector with integer numbers denoting group allocations

`A_c` an `KxNxC` array of starting values for the local parameter $\mathbf{A}_c$.

`Sigma_c` an `NxNxC` array of starting values for the local parameter $\mathbf{\Sigma}_c$.

A_g an KxNxG array of starting values for the group parameter $\mathbf{A}_g$.

Sigma_g an NxNxG array of starting values for the group parameter $\mathbf{\Sigma}_g$.

V an KxK matrix of starting values for the global parameter $\mathbf{V}$.

nu a positive scalar with starting values for the global parameter ν.

m a positive scalar with starting values for the global hyper-parameter m.

w a positive scalar with starting values for the global hyper-parameter w.

s a positive scalar with starting values for the global hyper-parameter s.

Methods

Public methods:

- StartingValuesBVARGROUPPRIORPANEL$new()
- StartingValuesBVARGROUPPRIORPANEL$get_starting_values()
- StartingValuesBVARGROUPPRIORPANEL$set_starting_values()
- StartingValuesBVARGROUPPRIORPANEL$clone()

StartingValuesBVARGROUPPRIORPANEL$new(): Create new starting values StartingValuesBVARGROUPPRIORPANEL

Usage:

```
StartingValuesBVARGROUPPRIORPANEL$new(
  group_allocation = 1:C,
  C,
  G = C,
  N,
  p,
  d = 0
)
```

Arguments:

group_allocation a numeric vector with integer numbers denoting group allocations

C a positive integer - the number of countries in the data.

G a positive integer specifying the number of country groups.

N a positive integer - the number of dependent variables in the model.

p a positive integer - the autoregressive lag order of the SVAR model.

d a positive integer - the number of exogenous variables in the model.

Returns: Starting values StartingValuesBVARGROUPPRIORPANEL

Examples:

```
# starting values for Bayesian Panel VAR 2-country model with 4 lags for a 3-variable system.
sv = specify_starting_values_bvarGroupPriorPANEL$new(C = 2, N = 3, p = 1)
```

StartingValuesBVARGROUPPRIORPANEL$get_starting_values(): Returns the elements of the starting values StartingValuesBVARGROUPPRIORPANEL as a list.

Usage:

```
StartingValuesBVARGROUPPRIORPANEL$get_starting_values()
```

Examples:

```
# starting values for a homoskedastic bsvar with 1 lag for a 3-variable system
sv = specify_starting_values_bvarGroupPriorPANEL$new(rep(1,2), C = 2, N = 3, p = 1)
sv$get_starting_values()   # show starting values as list
```

`StartingValuesBVARGROUPPRIORPANEL$set_starting_values()`: Returns the elements of the starting values StartingValuesBVARGROUPPRIORPANEL as a `list`.

Usage:

```
StartingValuesBVARGROUPPRIORPANEL$set_starting_values(last_draw)
```

Arguments:

`last_draw` a list containing the same elements as object StartingValuesBVARGROUPPRIORPANEL

Returns: An object of class StartingValuesBVARGROUPPRIORPANEL including the last draw of the current MCMC as the starting value to be passed to the continuation of the MCMC estimation.

Examples:

```
sv = specify_starting_values_bvarGroupPriorPANEL$new(rep(1,2), C = 2, N = 3, p = 1)

# Modify the starting values by:
sv_list = sv$get_starting_values()   # getting them as list
sv_list$A <- matrix(rnorm(12), 3, 4) # modifying the entry
sv$set_starting_values(sv_list)      # providing to the class object
```

`StartingValuesBVARGROUPPRIORPANEL$clone()`: The objects of this class are cloneable with this method.

Usage:

```
StartingValuesBVARGROUPPRIORPANEL$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
# starting values for a Bayesian Panel VAR
sv = specify_starting_values_bvarGroupPriorPANEL$new(rep(1,2), C = 2, G = 1, N = 3, p = 1)


## ------------------------------------------------
## Method `StartingValuesBVARGROUPPRIORPANEL$new()`
## ------------------------------------------------

# starting values for Bayesian Panel VAR 2-country model with 4 lags for a 3-variable system.
sv = specify_starting_values_bvarGroupPriorPANEL$new(C = 2, N = 3, p = 1)


## ------------------------------------------------
## Method `StartingValuesBVARGROUPPRIORPANEL$get_starting_values()`
## ------------------------------------------------
```

```
# starting values for a homoskedastic bsvar with 1 lag for a 3-variable system
sv = specify_starting_values_bvarGroupPriorPANEL$new(rep(1,2), C = 2, N = 3, p = 1)
sv$get_starting_values()   # show starting values as list


## ------------------------------------------------
## Method `StartingValuesBVARGROUPPRIORPANEL$set_starting_values()`
## ------------------------------------------------

sv = specify_starting_values_bvarGroupPriorPANEL$new(rep(1,2), C = 2, N = 3, p = 1)

# Modify the starting values by:
sv_list = sv$get_starting_values()   # getting them as list
sv_list$A <- matrix(rnorm(12), 3, 4) # modifying the entry
sv$set_starting_values(sv_list)      # providing to the class object
```

---

specify_starting_values_bvarPANEL

R6 Class Representing StartingValuesBVARPANEL

Description

The class StartingValuesBVARPANEL presents starting values for the Bayesian hierarchical panel VAR model.

Public fields

`A_c` an KxNxC array of starting values for the local parameter $\mathbf{A}_c$.

`Sigma_c` an NxNxC array of starting values for the local parameter $\mathbf{\Sigma}_c$.

`A` an KxN matrix of starting values for the global parameter $\mathbf{A}$.

`V` an KxK matrix of starting values for the global parameter $\mathbf{V}$.

`Sigma` an NxN matrix of starting values for the global parameter $\mathbf{\Sigma}$.

`nu` a positive scalar with starting values for the global parameter ν.

`m` a positive scalar with starting values for the global hyper-parameter m.

`w` a positive scalar with starting values for the global hyper-parameter w.

`s` a positive scalar with starting values for the global hyper-parameter s.

Methods

Public methods:

- `StartingValuesBVARPANEL$new()`
- `StartingValuesBVARPANEL$get_starting_values()`
- `StartingValuesBVARPANEL$set_starting_values()`

- StartingValuesBVARPANEL$clone()

`StartingValuesBVARPANEL$new()`: Create new starting values StartingValuesBVARPANEL

Usage:

```
StartingValuesBVARPANEL$new(C, N, p, d = 0)
```

Arguments:

`C` a positive integer - the number of countries in the data.

`N` a positive integer - the number of dependent variables in the model.

`p` a positive integer - the autoregressive lag order of the SVAR model.

`d` a positive integer - the number of `exogenous` variables in the model.

Returns: Starting values StartingValuesBVARPANEL

Examples:

```
# starting values for Bayesian Panel VAR 2-country model with 4 lags for a 3-variable system.
sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 4)
```

`StartingValuesBVARPANEL$get_starting_values()`: Returns the elements of the starting values StartingValuesBVARPANEL as a `list`.

Usage:

```
StartingValuesBVARPANEL$get_starting_values()
```

Examples:

```
# starting values for a homoskedastic bsvar with 1 lag for a 3-variable system
sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 1)
sv$get_starting_values()   # show starting values as list
```

`StartingValuesBVARPANEL$set_starting_values()`: Returns the elements of the starting values StartingValuesBVARPANEL as a `list`.

Usage:

```
StartingValuesBVARPANEL$set_starting_values(last_draw)
```

Arguments:

`last_draw` a list containing the same elements as object StartingValuesBVARPANEL.

Returns: An object of class StartingValuesBVARPANEL including the last draw of the current MCMC as the starting value to be passed to the continuation of the MCMC estimation.

Examples:

```
sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 1)

# Modify the starting values by:
sv_list = sv$get_starting_values()   # getting them as list
sv_list$A <- matrix(rnorm(12), 3, 4) # modifying the entry
sv$set_starting_values(sv_list)      # providing to the class object
```

`StartingValuesBVARPANEL$clone()`: The objects of this class are cloneable with this method.

Usage:

```
StartingValuesBVARPANEL$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
# starting values for a Bayesian Panel VAR
sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 1)


## ------------------------------------------------
## Method `StartingValuesBVARPANEL$new()`
## ------------------------------------------------

# starting values for Bayesian Panel VAR 2-country model with 4 lags for a 3-variable system.
sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 4)


## ------------------------------------------------
## Method `StartingValuesBVARPANEL$get_starting_values()`
## ------------------------------------------------

# starting values for a homoskedastic bsvar with 1 lag for a 3-variable system
sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 1)
sv$get_starting_values()   # show starting values as list


## ------------------------------------------------
## Method `StartingValuesBVARPANEL$set_starting_values()`
## ------------------------------------------------

sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 1)

# Modify the starting values by:
sv_list = sv$get_starting_values()   # getting them as list
sv_list$A <- matrix(rnorm(12), 3, 4) # modifying the entry
sv$set_starting_values(sv_list)      # providing to the class object
```

`specify_starting_values_bvars` *R6 Class Representing StartingValuesBVARs*

Description

The class StartingValuesBVARs presents starting values for the Bayesian hierarchical panel VAR model.

Public fields

`A_c` an `KxNxC` array of starting values for the local parameter $\mathbf{A}_c$.

`Sigma_c` an `NxNxC` array of starting values for the local parameter $\mathbf{\Sigma}_c$.

`nu` a `C`-vector of positive starting values for the parameter ν.

m a C-vector of starting values for the parameter m.

w a C-vector of positive starting values for the parameter w.

s a C-vector of positive starting values for the parameter s.

Methods

Public methods:

- StartingValuesBVARs$new()
- StartingValuesBVARs$get_starting_values()
- StartingValuesBVARs$set_starting_values()
- StartingValuesBVARs$clone()

StartingValuesBVARs$new(): Create new starting values StartingValuesBVARs

Usage:

```
StartingValuesBVARs$new(C, N, p, d = 0)
```

Arguments:

C a positive integer - the number of countries in the data.

N a positive integer - the number of dependent variables in the model.

p a positive integer - the autoregressive lag order of the SVAR model.

d a positive integer - the number of exogenous variables in the model.

Returns: Starting values StartingValuesBVARs

Examples:

```
# starting values for Bayesian VARs 2-country model with 4 lags for a 3-variable system.
sv = specify_starting_values_bvars$new(C = 2, N = 3, p = 4)
```

StartingValuesBVARs$get_starting_values(): Returns the elements of the starting values StartingValuesBVARs as a list.

Usage:

```
StartingValuesBVARs$get_starting_values()
```

Examples:

```
# starting values for bvars with 1 lag for a 3-variable system
sv = specify_starting_values_bvars$new(C = 2, N = 3, p = 1)
sv$get_starting_values()   # show starting values as list
```

StartingValuesBVARs$set_starting_values(): Returns the elements of the starting values StartingValuesBVARs as a list.

Usage:

```
StartingValuesBVARs$set_starting_values(last_draw)
```

Arguments:

last_draw a list containing the same elements as object StartingValuesBVARs.

Returns: An object of class StartingValuesBVARs including the last draw of the current MCMC as the starting value to be passed to the continuation of the MCMC estimation.

Examples:

```
sv = specify_starting_values_bvars$new(C = 2, N = 3, p = 1)

# Modify the starting values by:
sv_list = sv$get_starting_values()   # getting them as list
sv_list$A <- matrix(rnorm(12), 3, 4) # modifying the entry
sv$set_starting_values(sv_list)      # providing to the class object
```

`StartingValuesBVARs$clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
StartingValuesBVARs$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
# starting values for a Bayesian Panel VAR
sv = specify_starting_values_bvars$new(C = 2, N = 3, p = 1)


## ------------------------------------------------
## Method `StartingValuesBVARs$new()`
## ------------------------------------------------

# starting values for Bayesian VARs 2-country model with 4 lags for a 3-variable system.
sv = specify_starting_values_bvars$new(C = 2, N = 3, p = 4)


## ------------------------------------------------
## Method `StartingValuesBVARs$get_starting_values()`
## ------------------------------------------------

# starting values for bvars with 1 lag for a 3-variable system
sv = specify_starting_values_bvars$new(C = 2, N = 3, p = 1)
sv$get_starting_values()   # show starting values as list


## ------------------------------------------------
## Method `StartingValuesBVARs$set_starting_values()`
## ------------------------------------------------

sv = specify_starting_values_bvars$new(C = 2, N = 3, p = 1)

# Modify the starting values by:
sv_list = sv$get_starting_values()   # getting them as list
sv_list$A <- matrix(rnorm(12), 3, 4) # modifying the entry
sv$set_starting_values(sv_list)      # providing to the class object
```

summary.ForecastsPANEL *Provides posterior summary of country-specific Forecasts*

Description

Provides posterior summary of the forecasts including their mean, standard deviations, as well as 5 and 95 percentiles.

Usage

```
## S3 method for class 'ForecastsPANEL'
summary(object, which_c, ...)
```

Arguments

`object`	an object of class `ForecastsPANEL` obtained using the `forecast()` function containing draws the predictive density.
`which_c`	a positive integer or a character string specifying the country for which the forecast should be plotted.
`...`	additional arguments affecting the summary produced.

Value

A list reporting the posterior mean, standard deviations, as well as 5 and 95 percentiles of the forecasts for each of the variables and forecast horizons.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

See Also

forecast.PosteriorBVARPANEL, plot

Examples

```
# specify the model
specification = specify_bvarPANEL$new(
      ilo_dynamic_panel[1:5],
      exogenous = ilo_exogenous_variables[1:5])
burn_in       = estimate(specification, 5)            # run the burn-in
posterior     = estimate(burn_in, 5)                  # estimate the model

# forecast 3 years ahead
predictive    = forecast(
      posterior,
      3,
```

```
      exogenous_forecast = ilo_exogenous_forecasts[1:5])
summary(predictive, which_c = "ARG")
```

---

summary.PosteriorBVARGROUPPANEL
Provides posterior estimation summary for Bayesian Hierarchical Panel Vector Autoregressions

---

Description

Provides posterior mean, standard deviations, as well as 5 and 95 percentiles of the parameters for all C countries.

Usage

```
## S3 method for class 'PosteriorBVARGROUPPANEL'
summary(object, ...)
```

Arguments

object	an object of class PosteriorBVARGROUPPANEL obtained using the estimate() function applied to Vector Autoregressions containing draws from the posterior distribution of the parameters.
...	additional arguments affecting the summary produced.

Value

A list reporting the posterior mean, standard deviations, as well as 5 and 95 percentiles of the country-specific parameters.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

See Also

estimate.BVARGROUPPANEL, specify_bvarGroupPANEL

Examples

```
# specify the model
specification = specify_bvarGroupPANEL$new(
      data = ilo_dynamic_panel[1:5],
      exogenous = ilo_exogenous_variables[1:5],
      group_allocation = country_grouping_region[1:5]
)
burn_in       = estimate(specification, 5)            # run the burn-in
```

```
posterior     = estimate(burn_in, 5)                  # estimate the model
summary(posterior)
```

---

summary.PosteriorBVARGROUPPRIORPANEL

Provides posterior estimation summary for Bayesian Hierarchical Panel Vector Autoregressions with group-specific global prior

---

Description

Provides posterior mean, standard deviations, as well as 5 and 95 percentiles of the parameters for all C countries.

Usage

```
## S3 method for class 'PosteriorBVARGROUPPRIORPANEL'
summary(object, ...)
```

Arguments

object	an object of class PosteriorBVARGROUPPRIORPANEL obtained using the estimate() function applied to Vector Autoregressions containing draws from the posterior distribution of the parameters.
...	additional arguments affecting the summary produced.

Value

A list reporting the posterior mean, standard deviations, as well as 5 and 95 percentiles of the country-specific parameters.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

See Also

estimate.BVARGROUPPRIORPANEL, specify_bvarGroupPriorPANEL

Examples

```
# specify the model
specification = specify_bvarGroupPriorPANEL$new(
      data = ilo_dynamic_panel[1:5],
      group_allocation = country_grouping_region[1:5]
)
burn_in       = estimate(specification, 5)            # run the burn-in
posterior     = estimate(burn_in, 5)                  # estimate the model
summary(posterior)
```

summary.PosteriorBVARPANEL
Provides posterior estimation summary for Bayesian Hierarchical Panel Vector Autoregressions

Description

Provides posterior mean, standard deviations, as well as 5 and 95 percentiles of the parameters for all `C` countries.

Usage

```
## S3 method for class 'PosteriorBVARPANEL'
summary(object, ...)
```

Arguments

`object`	an object of class `PosteriorBVARPANEL` obtained using the `estimate()` function applied to Vector Autoregressions containing draws from the posterior distribution of the parameters.
`...`	additional arguments affecting the summary produced.

Value

A list reporting the posterior mean, standard deviations, as well as 5 and 95 percentiles of the country-specific parameters.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

See Also

`estimate.BVARPANEL`, `specify_bvarPANEL`

Examples

```
# specify the model
specification = specify_bvarPANEL$new(
      ilo_dynamic_panel[1:5],
      exogenous = ilo_exogenous_variables[1:5])
burn_in       = estimate(specification, 5)            # run the burn-in
posterior     = estimate(burn_in, 5)                  # estimate the model
summary(posterior)
```

`summary.PosteriorBVARs` *Provides posterior estimation summary for Bayesian Vector Autoregressions for dynamic panel data*

Description

Provides posterior mean, standard deviations, as well as 5 and 95 percentiles of the parameters for all `C` countries.

Usage

```
## S3 method for class 'PosteriorBVARs'
summary(object, ...)
```

Arguments

`object`	an object of class `PosteriorBVARs` obtained using the `estimate()` function applied to Vector Autoregressions containing draws from the posterior distribution of the parameters.
`...`	additional arguments affecting the summary produced.

Value

A list reporting the posterior mean, standard deviations, as well as 5 and 95 percentiles of the country-specific parameters.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

See Also

`estimate.BVARs`, `specify_bvars`

Examples

```
# specify the model
specification = specify_bvarPANEL$new(
      ilo_dynamic_panel[1:5],
      exogenous = ilo_exogenous_variables[1:5])
burn_in       = estimate(specification, 5)            # run the burn-in
posterior     = estimate(burn_in, 5)                  # estimate the model
summary(posterior)
```

summary.PosteriorFEVDPANEL

Provides posterior summary of forecast error variance decompositions

Description

Provides posterior means of the forecast error variance decompositions of each variable at all horizons.

Usage

```
## S3 method for class 'PosteriorFEVDPANEL'
summary(object, which_c, ...)
```

Arguments

`object`	an object of class `PosteriorFEVDPANEL` obtained using the `compute_variance_decompositions()` function containing draws from the posterior distribution of the forecast error variance decompositions.
`which_c`	a positive integer or a character string specifying the country for which the forecast should be plotted.
`...`	additional arguments affecting the summary produced.

Value

A list reporting the posterior mean of the forecast error variance decompositions of each variable at all horizons.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

See Also

`compute_variance_decompositions.PosteriorBVARPANEL`, `plot`

Examples

```
# specify the model and set seed
specification  = specify_bvarPANEL$new(ilo_dynamic_panel[1:5], p = 1)

# run the burn-in
burn_in        = estimate(specification, 5)

# estimate the model
posterior      = estimate(burn_in, 5)

# compute forecast error variance decomposition 4 years ahead
```

```
fevd           = compute_variance_decompositions(posterior, horizon = 4)
summary(fevd, which_c = "ARG")
```

Index