Package ‘BGVAR’

June 24, 2020

Type Package

Title Bayesian Global Vector Autoregressions

Version 2.0.1

Author Maximilian Boeck [aut, cre] (<https://orcid.org/0000-0001-6024-8305>),
Martin Feldkircher [aut] (<https://orcid.org/0000-0002-5511-9215>),
Florian Huber [aut] (<https://orcid.org/0000-0002-2896-7921>),
Christopher Sims [ctb]

Maintainer Maximilian Boeck <maximilian.boeck@wu.ac.at>

Description Estimation of Bayesian Global Vector Autoregressions (BGVAR) with different prior setups and the possibility to introduce stochastic volatility. Built-in priors include the Minnesota, the stochastic search variable selection and Normal-Gamma (NG) prior. For a reference see also Crespo Cuaresma, J., Feldkircher, M. and F. Huber (2016) ``Forecasting with Global Vector Autoregressive Models: a Bayesian Approach'', Journal of Applied Econometrics, Vol. 31(7), pp. 1371-1391 <doi:10.1002/jae.2504>. Post-processing functions allow for doing predictions, structurally identify the model with short-run or sign-restrictions and compute impulse response functions, historical decompositions and forecast error variance decompositions. Plotting functions are also available.

Encoding UTF-8

License GPL-3

Language en-US

Depends R (>= 2.10)

Imports abind, bayesm, coda, doParallel, foreach, GIGrvg, graphics, knitr, MASS, Matrix, methods, parallel, Rcpp (>= 1.0.3), stats, stochvol, utils, xts, zoo

Suggests testthat (>= 2.1.0), rmarkdown

LazyData true

LinkingTo Rcpp, RcppArmadillo, RcppProgress, stochvol, GIGrvg

RoxygenNote 7.1.0

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-06-24 12:10:14 UTC

R topics documented:

AIC.bgvar 3
avg.pair.cc 4
bgvar 5
BIC.bgvar 12
coef.bgvar 13
cond.pred 14
conv.diag 15
DIC 16
eerData 17
eerDataspf 18
fevd.decomp 19
fitted.bgvar 21
gfevd.decomp 21
hd.decomp 23
IRF 24
IRF.cf 30
list_to_matrix 31
logLik.bgvar 32
lps.bgvar.pred 33
matrix_to_list 33
monthlyData 34
pesaranData 35
plot.bgvar 36
plot.bgvar.fevd 37
plot.bgvar.irf 38
plot.bgvar.pred 39
plot.bgvar.resid 40
predict.bgvar 40
print.bgvar 41
print.bgvar.CD 42
print.bgvar.lps 43
print.bgvar.rmse 44
resid.corr.test 45
residuals.bgvar 46
rmse.bgvar.predict 47
summary.bgvar 48
vcov.bgvar 49

Index **51**

AIC.bgvar *Akaike Information Criterion*

Description

Computes the Akaike information criterion for an object `bgvar`.

Usage

```
## S3 method for class 'bgvar'
AIC(object, ..., k = 2)
```

Arguments

`object`	an object of class `bgvar`.
`...`	additional arguments.
`k`	the penalty per parameter to be used. Default is set to `k=2`.

Value

Returns a numeric value with the corresponding AIC.

Author(s)

Maximilian Boeck

References

Akaike, H. (1973) Information theory and an extension of the maximum likelihood principle. In: B. N. Petro and F. Csaki (eds.), 2nd International Symposium on Information Theory, pp. 267-281.

Akaike, H. (1974) A new look at the statistical model identification. *IEEE Transactions on Automatic Control AC-19*, pp. 716-723.

Examples

```
library(BGVAR)
data(eerData)
model.mn <- bgvar(Data=eerData,W=W.trade0012,plag=2,saves=100,burns=100,prior="MN")
AIC(model.mn)
```

`avg.pair.cc` *Average pairwise cross-sectional correlations*

Description

Computes average pairwise cross-sectional correlations of the data and the country models' residuals.

Usage

```
avg.pair.cc(obj, digits=3)
```

Arguments

`obj`	Either an object of class `bgvar` or residuals of class `bgvar.res`.
`digits`	Number of digits that should be used to print output to the console.

Details

If used for analyzing the country models' residuals, `avg.pair.cc` computes for each country and a given variable, the average cross-sectional correlation (either for the data or for the residuals). In theory, including foreign variables should soak up cross-sectional residual dependence and correlation of the residuals should be small. Otherwise dynamic analysis, especially using GIRFs, might lead to invalid results. See Dees et al. (2007) for more details.

Value

Returns a list with the following elements

`data.cor`	is a matrix containing in the rows the cross-sections and in the columns the cross-sectional pairwise correlations of the data per variable.
`resid.cor`	is a matrix containing in the rows the cross-sections and in the columns the cross-sectional pairwise correlations of the country models' residuals per variable.
`resid.corG`	is a matrix containing in the rows the cross-sections and in the columns the cross-sectional pairwise correlations of the global models' residuals per variable. Only available when `avg.pair.cc` has been applied to a `bgvar.res` object from `residuals`.
`data.res`	is a summary object showing the number and percentage of correlations <0.1, between 0.1-0.2, 0.2-0.5 and <0.5 per variable of the data.
`res.res`	is a summary object showing the number and percentage of correlations <0.1, between 0.1-0.2, 0.2-0.5 and <0.5 per variable of the country models' residuals. This is also what is used by `print.bgvar`.
`res.resG`	is a summary object showing the number and percentage of correlations <0.1, between 0.1-0.2, 0.2-0.5 and <0.5 per variable of the global models' residuals. Only available when `avg.pair.cc` has been applied to a `bgvar. res` object from `residuals`.

Author(s)

Martin Feldkircher

References

Dees, S., Di Mauro F., Pesaran, M. H. and Smith, L. V. (2007) *Exploring the international linkages of the euro area: A global VAR analysis.* Journal of Applied Econometrics, Vol. 22, pp. 1-38.

See Also

`bgvar` for estimation of a bgvar object. `residuals` for calculating the residuals from a `bgvar` object and creating a `bgvar.res` object.

Examples

```
library(BGVAR)
data(eerData)
model.mn <- bgvar(Data=eerData,W=W.trade0012,plag=1,SV=TRUE,
                  saves=100,burns=100,prior="MN")
avg.pair.cc(model.mn)

res <- residuals(model.mn)
avg.pair.cc(res)
```

`bgvar` *BGVAR*

Description

Estimates a Bayesian GVAR with either the Stochastic Search Variable Selection (SSVS), the Minnesota prior (MN), or the Normal-Gamma prior. All specifications can be estimated with stochastic volatility.

Usage

```
bgvar(Data, W, plag=1, saves=5000, burns=5000, prior="NG", SV=TRUE, h=0, thin=1,
    hyperpara=NULL, eigen=FALSE, variable.list=NULL, OE.weights=NULL, Wex.restr=NULL,
    Ex=NULL, trend=FALSE, save.country.store=FALSE, multithread=FALSE, verbose=TRUE)
```

Arguments

`Data`	Either a

- `list object` of length `N` that contains the data. Each element of the list refers to a country/entity. The number of columns (i.e., variables) in each country model can be different. The `T` rows (i.e., number of time observations), however, need to be the same for each country. Country and variable names are not allowed to contain a dot `.` (i.e., a dot) since this is our naming convention.
- `matrix object` of dimension `T` times `K`, with `K` denoting the sum of all endogenous variables of the system. The column names should consist of two parts, separated by a `.` (i.e., a dot). The first part should denote the country / entity name and the second part the name of the variable. Country and variable names are not allowed to contain a `.` (i.e., a dot).

`W` An N times N weight matrix with 0 elements on the diagonal and row sums that sum up to unity or a list of weight matrices.

`plag` Number of lags used (the same for domestic, exogenous and weakly exogenous variables.). Default set to `plag=1`.

`saves` Number of draws saved. Default set to `saves=5000`.

`burns` Number of burn-ins. Default set to `burns=5000`.

`prior` Either "SSVS", "MN" or "NG". See Details below.

`SV` If set to `TRUE`, models are fitted with stochastic volatility using the `stochvol` package. Due to storage issues, not the whole history of the `T` variance covariance matrices are kept, only the median. Consequently, the `BGVAR` package shows only one set of impulse responses (with variance covariance matrix based on mean sample point volatilities) instead of `T` sets. Specify `SV=FALSE` to turn SV off.

`h` Defines the hold-out sample. Default without hold-out sample, thus set to zero.

`thin` Is a thinning interval of the MCMC chain. As a rule of thumb, workspaces get large if saves/thin>500. Default set to `thin=1`.

`hyperpara` Is a list object that defines the hyperparameters when the prior is set to either `MN`, `SSVS` or `NG`.

- `a_1` is the prior hyperparameter for the inverted gamma prior (shape) (set a_1 = b_1 to a small value for the standard uninformative prior). Default is set to `a_1=0.01`.
- `b_1` is the prior hyperparameter for the inverted gamma prior (rate). Default is set to `b_1=0.01`.
- `prmean` Prior mean on the first lag of the autoregressive coefficients, standard value is `prmean=1` for non-stationary data. Prior mean for the remaining autoregressive coefficients automatically set to 0.
- `bmu` If `SV=TRUE`, this is the prior hyperparameter for the mean of the the mean of the log-volatilities. Default is `bmu=0`.
- `Bmu` If `SV=TRUE`, this is the prior hyperparameter for the variance of the mean of the log-volatilities. Default is `Bmu=100`.
- `a0` If `SV=TRUE`, this is the hyperparameter of the shape1 parameter for the Beta prior on the persistence parameter of the log-volatilities. Default is `a0=25`.

- b0 If SV=TRUE, this is the hyperparameter of the shape2 parameter for the Beta prior on the persistence parameter of the log-volatilities. Default is b0=1.5.
- Bsigma If SV=TRUE, this is the hyperparameter for the Gamma prior on the variance of the log-volatilities. Default is set to Bsigma=1.
- "MN"
  - shrink1 Starting value of shrink1. Default set to 0.1.
  - shrink2 Starting value of shrink2. Default set to 0.2.
  - shrink3 Hyperparameter of shrink3. Default set to 100.
  - shrink4 Starting value of shrink4. Default set to 0.1.
- "SSVS"
  - tau0 is the prior variance associated with the normal prior on the regression coefficients if a variable is NOT included (spike, tau0 should be close to zero).
  - tau1 is the prior variance associated with the normal prior on the regression coefficients if a variable is included (slab, tau1 should be large).
  - kappa0 is the prior variance associated with the normal prior on the covariances if a covariance equals zero (spike, kappa0 should be close to zero).
  - kappa1 is the prior variance associated with the normal prior on the covariances if a covariance is unequal to zero (slab, kappa1 should be large).
  - p_i is the prior inclusion probability for each regression coefficient whether it is included in the model (default set to p_i=0.5).
  - q_ij is the prior inclusion probability for each covariance whether it is included in the model (default set to q_ij=0.5).
- "NG":
  - e_lambda Prior hyperparameter for the Gamma prior on the lag-specific shrinkage components, standard value is e_lambda=1.5.
  - d_lambda Prior hyperparameter for the Gamma prior on the lag-specific shrinkage components, standard value is d_lambda=1.
  - a_start Parameter of the Normal-Gamma prior that governs the heaviness of the tails of the prior distribution. A value of a_start=1 would lead to the Bayesian LASSO. Default value differs per entity and set to a_start=1/log(M), where M is the number of endogenous variables per entity.
  - sample_A If set to TRUE a_start is sampled.

eigen Set to TRUE if you want to compute the largest eigenvalue of the companion matrix for each posterior draw. If the modulus of the eigenvalue is significantly larger than unity, the model is unstable. Unstable draws exceeding an eigenvalue of one are then excluded. If eigen is set to a numeric value, then this corresponds to the maximum eigenvalue. The default is set to 1.05 (which excludes all posterior draws for which the eigenvalue of the companion matrix was larger than 1.05 in modulus).

variable.list
: In case W is a list of weight matrices, specify here which set of variables should be weighted by which weight matrix. See the help file on getweights for details. Default is NULL.

OE.weights
: Default value is set to NULL. Can be used to provide information of how to handle additional country models (other entities). Additional country models can be used to endogenously determine variables that are (weakly) exogenous for the majority of the other country models. As examples, one could think of an additional oil price model (see also Mohaddes and Raissi 2019) or a model for the joint euro area monetary policy (see also Georgiadis 2015; Feldkircher, Gruber and Huber (2020)). The data for these additional country models has to be contained in Data. The number of additional country models is unlimited. Each list entry of OE.weights has to be named similar to the name of the additional country model contained in Data. Each slot of OE.weight has to contain the following information:

  - weights a vector of weights with names relating to the countries for which data should be aggregated. Can also relate to a subset of countries contained in the data.
  - variables a vector of variables names that should be included in the additional country model. Variables that are not contained in the data slot of the extra country model are assumed to be weakly exogenous for the additional country model (aggregated with weight).
  - exo a vector of variable names that should be fed into the other countries as (weakly) exogenous variables.

Wex.restr
: A character vector that contains variables that should only be specified as weakly exogenous if not contained as endogenous variable in a particular country. An example that has often been used in the literature is to place these restrictions on nominal exchange rates. Default is NULL in which case all weakly exogenous variables are treated symmetrically. See function getweights for more details.

Ex
: For including truly exogenous variables to the model. Either a

  - list object of maximum length N that contains the data. Each element of the list refers to a country/entity and has to match the country/entity names in Data. If no truly exogenous variables are added to the respective country/entity model, omit the entry. The T rows (i.e., number of time observations), however, need to be the same for each country. Country and variable names are not allowed to contain a dot . (i.e., a dot) since this is our naming convention.
  - matrix object of dimension T times number of truly exogenous variables. The column names should consist of two parts, separated by a . (i.e., a dot). The first part should denote the country / entity name and the second part the name of the variable. Country and variable names are not allowed to contain a . (i.e., a dot).

trend
: If set to TRUE a deterministic trend is added to the country models.

save.country.store
: If set to TRUE then function also returns the container of all draws of the individual country models. Significantly raises object size of output and default is thus set to FALSE.

multithread — If set to TRUE parallel computing using the packages foreach and doParallel. Number of cores is set to maximum number of cores in the computer. This option is recommended when working with sign restrictions to speed up computations. Default is set to FALSE and thus no parallelization.

verbose — If set to FALSE it suppresses printing messages to the console.

Details

We provide three priors, the Minnesota labeled MN, the SSVS and the Normal-Gamma prior. The first one has been implemented for global VARs in Feldkircher and Huber (2016) and the second one in Crespo Cuaresma et al. (2016), while the last one has been introduced to VAR modeling in Huber and Feldkircher (2019). Please consult these references for more details on the specification. In the following we will briefly explain the difference between the three priors. The Minnesota prior pushes the variables in the country-specific VAR towards their unconditional stationary mean, or toward a situation where there is at least one unit root present. The SSVS prior is a form of a 'spike' and 'slab' prior. Variable selection is based on the probability of assigning the corresponding regression coefficient to the 'slab' component. If a regression coefficient is non informative, the 'spike' component pushes the associated posterior estimate more strongly towards zero. Otherwise, the slab component resembles a non-informative prior that has little impact on the posterior. Following George et. al. (2008) we set the prior variances for the normal distribution in a semi-automatic fashion. This implies scaling the mixture normal with the OLS standard errors of the coefficients for the full model. The NG prior is a form of global-local shrinkage prior. Hence, the local component shrinks each coefficient towards zero if there is no information for the associated dependent variable. Otherwise, the prior exerts a fat-tail structure such that deviations from zero are possible. The global component is present for each lag, thus capturing the idea that higher lags should be shrunk more aggressively towards zero.

Value

Returns a list of class bgvar with the following elements:

- args is a list object that contains the arguments submitted to function bgvar.
- xglobal is a matrix object of dimension T times N (T # of observations, K # of variables in the system).
- gW is the global weight matrix. It is a list, with N entries, each of which contains the weight matrix of each country.
- country.res is a matrix that contains the posterior mean of the country models' residuals. The residuals have been obtained as a running mean and thus always relate to the full set of posterior draws. This implies that in case you have opted for trimming the draws the residuals do not correspond to the posterior draws of the "trimmed" coefficients. This is a storage problem, rather than a statistical problem. Experiments, however, show that residual properties (autocorrelation, cross-sectional correlation) of trimmed and reported residuals are close.
- stacked results
  - S_large is a three-dimensional array (K times K times saves) of the (block-diagonal) posterior variance covariance matrix.
  - F_large is a four-dimensional array (K times K times lags times saves) of the coefficients.

  - Ginv_large is a three-dimensional array (K times K times saves) of the inverse of the G matrix.
  - A_large is a three-dimensional array (K times K+1 times saves) of the posterior estimates for the K coefficients plus a global constant.
  - F.eigen in case eigen="TRUE", returns a vector that contains for each posterior draw the modulus of the largest eigenvalue of the companion matrix.
  - trim.info is a character vector. Contains information regarding the nr. of stable draws out of total (thinned) draws. Experience shows that a maximum eigenvalue of 1.05 seems a reasonable choice when working with data in levels to generate stable impulse responses.
- cc.results each entry of this list contains an list object of length N. Each entry in the list corresponds to one country model and contains one of the following posterior medians.
  - coeffs contains in each entry the matrix with the posterior median of the estimated coefficients. Columns of the matrix correspond to an equation in the country model (i.e., the dependent variable) and rows to coefficient estimates of the explanatory variables.
  - sig contains in each entry the variance-covariance matrix for each point in time. If SV=FALSE all entries along the time dimension are the same.
  - theta contains in each entry the estimated prior variances for the coefficients. Explains how much shrinkage is induced on each coefficient depending on the prior setup.
  - res contains in each entry a matrix of dimension (T-p times K) with the posterior median of the residuals of the cross-country models.
  - shrink in case prior="MN" each entry contains the estimated shrinkage parameters.
  - PIP in case prior="SSVS" returns a list object. The first slot in the list PIP.cc, is a list of length N and contains the posterior inclusion probabilities of the country models. The second slot in the list, named PIP.avg yields simple averages (over the country models where a particular variable has been included) of the posterior inclusion probabilities.
  - lambda2 in case prior="NG" each entry contains the estimated global shrinkage parameters. It is a matrix of dimension (p+1 times 3). Columns refer to the endogenous, weakly exogenous and shrinkage parameters for the covariances. Rows correspond to different degree of shrinkage per lag of the variables starting with the contemporaneous lag (only for weakly exogenous variables). In case of the covariances just one global shrinkage parameter is estimated.
  - tau in case prior="NG" each entry contains the estimated parameter that governs the heaviness of the tails of the marginal prior distribution of the coefficients associated to endogenous variables. Structure is the same as lambda2.

Author(s)

Maximilian Boeck, Martin Feldkircher, Florian Huber

References

Crespo Cuaresma, J., Feldkircher, M. and F. Huber (2016) Forecasting with Global Vector Autoregressive Models: A Bayesian Approach. *Journal of Applied Econometrics*, Vol. 31(7), pp. 1371-1391.

Doan, T. R., Litterman, B. R. and C. A. Sims (1984) Forecasting and Conditional Projection Using Realistic Prior Distributions. *Econometric Reviews*, Vol. 3, pp. 1-100.

Dovern, J., Feldkircher, M. and F. Huber (2016) Does joint modelling of the world economy pay off? Evaluating multivariate forecasts from a Bayesian GVAR. *Journal of Economic Dynamics and Control*, Vol. 70, pp. 86-100.

Feldkircher, M. and F. Huber (2016) The International Transmission of US Shocks - Evidence from Bayesian Global Vector Autoregressions. *European Economic Review*, Vol. 81, pp. 167-188.

Feldkircher, M. Gruber, T. and F. Huber (2020) International effects of a compression of euro area yield curves. *Journal of Banking & Finance*, Vol. 113, pp. 11-14.

George, E.I., Sun, D. and S. Ni (2008) Bayesian stochastic search for var model restrictions. *Journal of Econometrics*, Vol. 142, pp. 553-580.

Georgiadis, G. (2015) Examining asymmetries in the transmission of monetary policy in the euro area: Evidence from a mixed cross-section global VAR model. *European Economic Review*, Vol. 75, pp. 195-215.

Huber, F. and M. Feldkircher (2016) Adaptive Shrinkage in Bayesian Vector Autoregressive Models. *Journal of Business and Economic Statistics*, Vol. 37(1), pp. 27-39.

Mohaddes, K. and M. Raissi (2018). Compilation, Revision and Updating of the Global VAR (GVAR) Database, 1979Q2-2016Q4. University of Cambridge: Faculty of Economics (mimeo).

Mohaddes, K. and M. Raissi (2019) The US oil supply revolution and the global economy. *Empirical Economics*, Vol. 57, pp. 515-546.

Pesaran, M.H., Schuermann T. and S.M. Weiner (2004) Modeling Regional Interdependencies Using a Global Error-Correcting Macroeconometric Model. *Journal of Business and Economic Statistics*, Vol. 22, pp. 129-162.

Sims, C. A. (1992) Bayesian Inference for Multivariate Time Series with Trend. *Mimeo*, presented at the American statistical Association meeting.

Sims, C.A. and T. Zha (1998) Bayesian Methods for Dynamic Multivariate Models. *International Economic Review*, Vol. 39, pp. 949-968.

Examples

```
library(BGVAR)
# replicate Feldkircher and Huber (2016) using trade based weights
data(eerData)
hyperpara <- list(tau0=0.1,tau1=3,kappa0=0.1,kappa1=7,a_1=0.01,b_1=0.01,p_i=0.5,q_ij=0.5)
model.ssvs <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=100,burns=100,
                    prior="SSVS",SV=FALSE,hyperpara=hyperpara,thin=1)
print(model.ssvs)

data("eerData")
variable.list<-list();variable.list$real<-c("y","Dp","tb");variable.list$fin<-c("stir","ltir","rer")
model.mn <- bgvar(Data=eerData, W=W.list[c("tradeW.0012","finW0711")], plag=1, saves=200,
                  burns=100,prior="MN",SV=TRUE,thin=2,variable.list=variable.list)
print(model.mn)

data(monthlyData)
EA.weights$variables <- c("EAstir","total.assets","M3","ciss","y","p")
OC.weights$variables <- c("poil","qoil","y")
```

```
OE.weights <- list(EB=EA.weights,OC=OC.weights)
hyperpara<-list(c_tau = 0.01, d_tau = 0.01,e_lambda=1.5,d_lambda=1,
                prmean=0,a_i=0.01,b_i=0.01,a_start=.6,sample_A=FALSE)
model.ssvs <- bgvar(Data=monthlyData,W=W,plag=2,saves=100,burns=100,prior="SSVS",
                    hyperpara=hyperpara,eigen=TRUE,SV=TRUE,OE.weights=OE.weights)
print(model.ssvs)
```

BIC.bgvar *Bayesian Information Criterion*

Description

Computes the Bayesian information criterion for an object bgvar.

Usage

```
## S3 method for class 'bgvar'
BIC(object, ...)
```

Arguments

object	an object of class bgvar.
...	additional arguments.

Value

Returns a numeric value with the corresponding BIC.

Author(s)

Maximilian Boeck

References

Schwartz, G. E. (1978) *Estimating the dimension of a model.* Annals of Statistics, Vol. 6(2), pp. 461-464.

Examples

```
library(BGVAR)
data(eerData)
model.mn <- bgvar(Data=eerData,W=W.trade0012,plag=2,saves=100,burns=100,prior="MN")
BIC(model.mn)
```

`coef.bgvar` *Extract model coefficients*

Description

Extracts the global model coefficients for `bgvar` for certain quantiles of the posterior distribution. `coefficients` is an *alias* for it.

Usage

```
## S3 method for class 'bgvar'
coef(object, ..., quantile = 0.5)

## S3 method for class 'bgvar'
coefficients(object, ..., quantile = 0.5)
```

Arguments

`object`	an object of class `bgvar`.
`...`	additional arguments.
`quantile`	reported quantiles. Default is set to the median.

Value

Returns an `q` times `K` times `K` times `p` array of the global coefficients, where `q` is the number of specified quantiles (this dimension is dropped if `q=1`), `K` the number of endogenous variables and `p` number of lags.

Examples

```
library(BGVAR)
data(eerData)
model.ng <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=100,burns=100)
coef(model.ng)


coefficients(model.ng)
```

cond.pred *Conditional Forecasts*

Description

Function that computes conditional forecasts for Bayesian Vector Autoregressions.

Usage

```
cond.predict(constr, bgvar.obj, pred.obj, constr_sd=NULL, verbose=TRUE)
```

Arguments

constr	a matrix containing the conditional forecasts of size horizon times K, where horizon corresponds to the forecast horizon specified in `pred.obj`, while K is the number of variables in the system. The ordering of the variables have to correspond the ordering of the variables in the system. Rest is just set to NA.
bgvar.obj	an item fitted by `bgvar`.
pred.obj	an item fitted by `predict`. Note that `save.store=TRUE` is required as argument!
constr_sd	a matrix containing the standard deviations around the conditional forecasts. Must have the same size as `constr`.
verbose	If set to `FALSE` it suppresses printing messages to the console.

Details

Conditional forecasts need a fully identified system. Therefore this function utilizes short-run restrictions via the Cholesky decomposition on the global solution of the variance-covariance matrix of the Bayesian GVAR.

Value

Returns an object of class `bgvar.pred` with the following elements

- `fcast` is a K times fhorz times 5-dimensional array that contains 16%th, 25%th, 50%th, 75%th and 84% percentiles of the conditional posterior predictive distribution.
- `xglobal` is a matrix object of dimension T times N (T # of observations, K # of variables in the system).
- `fhorz` specified forecast horizon.

Author(s)

Maximilian Boeck

References

Jarocinski, M. (2010) *Conditional forecasts and uncertainty about forecasts revisions in vector autoregressions.* Economics Letters, Vol. 108(3), pp. 257-259.

Waggoner, D., F. and T. Zha (1999) *Conditional Forecasts in Dynamic Multivariate Models.* Review of Economics and Statistics, Vol. 81(4), pp. 639-561.

Examples

```
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,prior="SSVS",
                      eigen=TRUE)

# compute predictions
fcast <- predict(model.ssvs.eer,fhorz=8,save.store=TRUE)

# set up constraints matrix of dimension fhorz times K
constr <- matrix(NA,nrow=fcast$fhorz,ncol=ncol(model.ssvs.eer$xglobal))
colnames(constr) <- colnames(model.ssvs.eer$xglobal)
constr[1:5,"US.Dp"] <- model.ssvs.eer$xglobal[76,"US.Dp"]

# add uncertainty to conditional forecasts
constr_sd <- matrix(NA,nrow=fcast$fhorz,ncol=ncol(model.ssvs.eer$xglobal))
colnames(constr_sd) <- colnames(model.ssvs.eer$xglobal)
constr_sd[1:5,"US.Dp"] <- 0.001

cond_fcast <- cond.predict(constr, model.ssvs.eer, fcast, constr_sd)
plot(cond_fcast, resp="US.Dp", Cut=10)
```

---

conv.diag *MCMC convergence diagnostics*

Description

This function computes Geweke's Convergence diagnostic making use of the coda package.

Usage

```
conv.diag(object, crit.val=1.96)
```

Arguments

object	a fitted bgvar object.
crit.val	critical value used for test statistic.

Details

Geweke (1992) proposed a convergence diagnostic for Markov chains based on a test for equality of the means of the first and last part of a Markov chain (by default we use the first 10% and the last 50%). If the samples are drawn from the stationary distribution of the chain, the two means are equal and Geweke's statistic has an asymptotically standard normal distribution. The test statistic is a standard Z-score: the difference between the two sample means divided by its estimated standard error. The standard error is estimated from the spectral density at zero and so takes into account any autocorrelation.

Value

Returns an object of class `bgvar.CD`. This is a list with

- `geweke.z` Z-scores for a test of equality of means between the first and last parts of the chain. A separate statistic is calculated for each variable in each chain.
- `perc` is the percentage of Z-scores exceeding `crit.val` (in absolute terms).

Author(s)

Martin Feldkircher

References

Geweke, J. (1992) Evaluating the accuracy of sampling-based approaches to calculating posterior moments. *Bayesian Statistics* 4 (edited by JM Bernado, JO Berger, AP Dawid and AFM Smith). Clarendon Press, Oxford, UK.

See Also

`geweke.diag` in the `coda` package.

Examples

```
library(BGVAR)
data(eerData)
model.mn <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=200,burns=200,prior="MN")
geweke <- conv.diag(model.mn)
```

DIC *Deviance Information Criterion*

Description

Computes the Deviance information criterion for an object `bgvar`.

Usage

```
DIC(object, ...)
```

Arguments

`object`	an object of class `bgvar`.
`...`	additional arguments.

Value

Returns a numeric value with the corresponding DIC.

Author(s)

Maximilian Boeck

References

Spiegelhalter, D. J. and Best, N. G., Carlin, B. P. and Linde, A. (2002) *Bayesian measures of model complexity and fit.* Journal of the Royal Statistical Society, Series B, Vol. 64(4), pp. 583-639.

Examples

```
set.seed(1)
library(BGVAR)
data(eerData)
model.mn <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=100,burns=100,prior="MN")
DIC(model.mn)
```

`eerData` *Example data set to replicate Feldkircher and Huber (2016)*

Description

This data set contains 76 quarterly observations by country, spanning the period from 1995Q1 to 2013Q4. The country coverage is 43 countries and the euro area (EA) as a regional aggregate.

Usage

```
eerData
```

Format

The data loads two objects `eerData`, which is a list object of length `N` (i.e, the number of countries) and `W.trade0012`, which is an `N` times `N` weight matrix with rowsums summing up to unity and zero elements on its diagonal. The global variable, oil prices, is included in the US country model as e.g., in Dees et al. (2007). The countries are abbreviated using ISO-2 codes. The weight matrix corresponds to average annual bilateral trade flows (including services) over the period from 2000 to 2012.`eerData` contains the country data, for more details, see below:

`W.trade0012` `N` times `N` weight matrix based on trade flows, rowsums equal unity.

`W.list` A list of 10 weight matrices, described in Feldkircher and Huber (2016).

`eerData` is a list object of length `N` containing

- `y` Real GDP, average of 2005=100. Seasonally adjusted, in logarithms.
- `Dp` Consumer prices (period-on-period). CPI seasonally adjusted, in logarithm.
- `stir` Short-term interest rate, typically 3-months money market rate.
- `ltir` Long-term interest rates, typically 10-year government bond yields.
- `reer` Real effective exchange rate, deflated by consumer prices.
- `tb` Trade balance (ratio of real exports to real imports).
- `poil` Price of oil, seasonally adjusted, in logarithms.

`eerDataspf` *eerData extended with expectations data*

Description

This data set contains 76 quarterly observations by country, spanning the period from 1995Q1 to 2013Q4. The country coverage is 43 countries + the euro area (EA) as a regional aggregate. Additionally, the US country dataset is extended with four quarter ahead expectation data on output, prices and short-term interest rates from the Survey of Professional Forecasters.

Usage

```
eerDataspf
```

Format

The data loads two objects `eerData`, which is a list object of length `N` (i.e, the number of countries) and `W.trade0012`, which is an `N` times `N` weight matrix with rowsums summing up to unity and zero elements on its diagonal. The global variable, oil prices, is included in the US country model as e.g., in Dees et al. (2007). The countries are abbreviated using ISO-2 codes. The weight matrix corresponds to average annual bilateral trade flows (including services) over the period from 2000 to 2012.`eerData` contains the country data, for more details, see below:

`W.trade0012spf` `N` times `N` weight matrix based on trade flows, rowsums equal unity.

`eerDataspf` is a list object of length `N` containing

- `y_t+4` four quarter ahead expectation of Real GDP growth.

- `Dp_t+4` four quarter ahead expectation of consumer price inflation.
- `stir_t+4` four quarter ahead expectation of short-term interest rates.
- `y` Real GDP growth.
- `Dp` Consumer price inflation (period-on-period).
- `stir` Short-term interest rate, typically 3-months money market rate.
- `ltir` Long-term interest rates, typically 10-year government bond yields.
- `reer` Real effective exchange rate, deflated by consumer prices.
- `tb` Trade balance (ratio of real exports to real imports).
- `poil` Price of oil, seasonally adjusted, in logarithms.

`fevd.decomp` *Forecast Error Variance Decomposition*

Description

This function calculates the forecast error variance decomposition (FEVDs) for Cholesky and sign-identified shocks.

Usage

```
fevd.decomp(obj, R=NULL, var.slct=NULL, verbose=TRUE)
```

Arguments

`obj`	an object of class `bgvar.irf`.
`R`	If `NULL` and the `obj` has been fitted via sign restrictions, the rotation matrix is used that minimizes the distance to the median impulse responses at the posterior median.
`var.slct`	character vector that contains the variables for which forecast error variance decomposition should be performed. If `NULL` the FEVD is computed for the whole system, which is very time consuming.
`verbose`	If set to `FALSE` it suppresses printing messages to the console.

Details

Since the calculations are very time consuming, the FEVDs are based on the posterior median only (as opposed to calculating FEVDs for each MCMC sweep). In case the underlying shock has been identified via sign restrictions, the rotation matrix corresponds to the one that fulfills the sign restrictions at the posterior median of the estimated coefficients. More precisely, the algorithm searches for 50 rotation matrices that fulfill the sign restrictions at the *posterior median* of the coefficients and then singles out the rotation matrix that minimizes the distance to the median of the impulse responses as suggested in Fry and Pagan (2011).

Value

Returns a list with two elements

- FEVD an array of size (K times horizon times N), where K are all variables in the system, horizon is the specified impulse response horizon and N is the size of the decomposed structural variables (if var.slct=NULL then K=N).
- xglobal used data of the model.

Author(s)

Maximilian Boeck, Martin Feldkircher, Florian Huber

See Also

IRF

Examples

```
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,
                      prior="SSVS",thin=1,eigen=TRUE)

# US monetary policy shock
shocks<-list();shocks$var="stir";shocks$cN<-"US";shocks$ident="chol";shocks$scal=-100
irf.chol.us.mp<-IRF(obj=model.ssvs.eer,shock=shocks,nhor=48)

# calculates FEVD for variables US.Dp and EA.y
fevd.us.mp=fevd.decomp(obj=irf.chol.us.mp,var.slct=c("US.Dp","EA.y"))

# US monetary policy shock with sign restrictions
sign.constr<-list()
sign.constr$shock1$shock             <- c("US.stir")
sign.constr$shock1$restrictions$res1 <- c("US.y")
sign.constr$shock1$restrictions$res2 <- c("US.Dp")
sign.constr$shock1$sign              <- c(">","<","<")
sign.constr$shock1$rest.horz         <- c(1,1,1)
sign.constr$shock1$constr            <- c(1,1,1)
sign.constr$shock1$scal              <- +100
sign.constr$MaxTries<-200
irf.sign.us.mp<-IRF(obj=model.ssvs.eer,sign.constr=sign.constr,nhor=24)

# calculates FEVD for variables US.Dp and EA.y
fevd.us.mp=fevd.decomp(obj=irf.sign.us.mp,var.slct=c("US.Dp","EA.y"))

# NOT RUN - calculates FEVDs for all variables in the system, very time consuming
## Not run:
```

`fitted.bgvar` *Extract Model Fitted Values*

Description

Extracts the fitted values for `bgvar`.

Usage

```
## S3 method for class 'bgvar'
fitted(object, ..., global = TRUE)
```

Arguments

`object`	an object of class `bgvar`.
`...`	additional arguments.
`global`	if `TRUE` global fitted values are returned otherwise country fitted values.

Value

Returns an `T` times `K` matrix, where `T` is the number of observations and `K` number of endogenous variables.

Examples

```
library(BGVAR)
data(eerData)
model.ng <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=100,burns=100)
fitted(model.ng)
```

`gfevd.decomp` *Generalized Forecast Error Variance Decomposition*

Description

This function calculates a complete generalized forecast error variance decomposition (GFEVDs) based on generalized impulse response functions akin to Lanne-Nyberg (2016). The Lanne-Nyberg (2016) corrected GFEVD sum up to unity.

Usage

```
gfevd.decomp(obj, nhor=24, running=TRUE, multithread=FALSE, verbose=TRUE)
```

Arguments

`obj`	an object of class `bgvar`.
`nhor`	the forecast horizon.
`running`	Default is set to `TRUE` and implies that only a running mean over the posterior draws is calculated. A full analysis including posterior bounds is likely to cause memory issues.
`multithread`	If set to `TRUE` parallel computing using the packages `foreach` and `doParallel`. Number of cores is set to maximum number of cores in the computer. This option is recommended when working with sign restrictions to speed up computations. Default is set to `FALSE` and thus no parallelization.
`verbose`	If set to `FALSE` it suppresses printing messages to the console.

Value

Returns a list with two elements

- `GFEVD` a three or four-dimensional array, with the first dimension referring to the K time series that are decomposed into contributions of K time series (second dimension) for `nhor` forecast horizons. In case `running=TRUE` only the posterior mean else also its 16% and 84% credible intervals is contained in the fourth dimension.
- `xglobal` used data of the model.

Author(s)

Maximilian Boeck, Martin Feldkircher

References

Lanne, M. and H. Nyberg (2016) *Generalized Forecast Error Variance Decomposition for Linear and Nonlinear Multivariate Models.* Oxford Bulletin of Economics and Statistics, Vol. 78(4), pp. 595-603.

See Also

`bgvar`.

Examples

```
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,
                      prior="SSVS",thin=1,eigen=TRUE)

# Calculates running mean GFEVDs for all variables in the system
GFEVD<-gfevd.decomp(model.ssvs.eer,nhor=24,running=TRUE)
```

`hd.decomp` *Historical Decomposition*

Description

A function that calculates historical decomposition (HD) of the time series and the structural error.

Usage

```
hd.decomp(obj, R=NULL, verbose=TRUE)
```

Arguments

`obj`	an item fitted by `IRF`.
`R`	If `NULL` and the `irf.bgvar` object has been fitted via sign restrictions, the rotation matrix is used that minimizes the distance to the median impulse responses at the posterior median.
`verbose`	If set to `FALSE` it suppresses printing messages to the console.

Details

To save computational time as well as due to storage limits, both functions are based on the posterior median (as opposed to calculating HDs and the structural error for each draw of the MCMC chain). In case the shock has been identified via sign restrictions, a rotation matrix has to be selected to calculate both statistics. If not specified otherwise (via `R`), the algorithm searches for 50 rotation matrices that fulfill the sign restrictions at the *posterior median* of the coefficients and then singles out the rotation matrix tha minimizes the distance to the median of the impulse responses as suggested in Fry and Pagan (2011).

Value

Returns a list with the following objects

- `hd_array` is a three-dimensional array with the first dimension referring to the K time series, the second to the T observations and the third dimensions containing the contribution of the shocks in explaining historically deviations in the time series from their trend. The third dimension is K+3, since the last three entries contain the contributions of the constant, the initial condition and a residual component that the contributions sum up to the original time series. If a trend i specified in the model the third dimension is K+3 with trend ordered after the constant.
- `struc.shcok` contains the structural shock.
- `x` is a matrix object that contains the original time series, which is of dimension K times (T-plag).

Author(s)

Maximilian Boeck, Martin Feldkircher, Florian Huber

References

Fry, R. and A. Pagan (2011) *Sign restrictions in Structural Vector Autoregressions: A Critical Review*. Journal of Economic Literature, Vol. 49(4), pp. 938-960.

See Also

bgvar and IRF.

Examples

```
set.seed(571)
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,
                      prior="SSVS",thin=1,eigen=TRUE)
# US monetary policy shock
shocks<-list();shocks$var="stir";shocks$cN<-"US";shocks$ident="chol";shocks$scal=-100
irf.chol.us.mp <- IRF(obj=model.ssvs.eer,shock=shocks,nhor=48)

HD <- hd.decomp(irf.chol.us.mp)
# summing them up should get you back the original time series
org.ts<-apply(HD$hd_array,c(1,2),sum)
matplot(cbind(HD$x[,1],org.ts[,1]),type="l",ylab="")
legend("bottomright",c("hd series","original"),col=c("black","red"),lty=c(1,2),bty="n")
```

IRF *Impulse Response Functions*

Description

This function calculates three alternative ways of dynamic responses, namely generalized impulse response functions (GIRFs) as in Pesaran and Shin (1998), orthogonalized impulse response functions using a Cholesky decomposition and finally impulse response functions given a set of user-specified sign restrictions.

Usage

```
IRF(obj, nhor=24, shock=NULL, sign.constr=NULL, save.store=FALSE, multithread=FALSE,
            verbose=TRUE)
```

Arguments

obj	an object of class bgvar.
nhor	forecasting horizon.

`shock` This is a list object. It should contain an entry labeled `var` that contains the name of the variable to be shocked. Also it should contain a list entry labeled `cN` that contains a character (or character vector) of the country (countries) in which the variable should be shocked. Finally it has to contain an entry labeled `ident` that is either `chol` if the shock is based on a short-run identification scheme done with the Cholesky decomposition or `girf` if generalized impulse responses should be calculated. In case impulses should be normalized (e.g., a +100bp increase in the interest rate), add another entry `scal` that contains a numeric value of the desired impact normalization.

`sign.constr` the user should submit a list containing the following entries

- `shock1` is a list object that defines sign restrictions for a particular shock.
  - `shockvar` is a character vector containing the country and variable to shock separated by a dot. Example, "AT.ltir" (long-term interest rates in Austria).
  - `restrictions` is a list containing the variables to restrict. Can have several sub-list restrictions, e.g., `sign.constr$shock1$restricionts$rest1=c("DE.y","AT.` `sign.constr$shock1$restricionts$rest2=c("NL.p","AT.p")`, putting restrictions on GDP in Germany and Austria and a second set of restrictions on prices in the Netherlands and Austria.
  - `sign` is a character vector of length of set of restrictions + 1, specifying the signs to impose. Use either `>`, `<` or `0`. The latter implements zero restrictions according to Arias et al. (2019). First entry is for the shock, say `AT.ltir` should go up, the following entries refer to the restrictions. `sign.constr$shock1$sign=c(">","<","<")` would impose `AT.ltir` to increase, and variables specified in `sign.constr$shock1$restriciונts$rest1` and `sign.constr$shock1$restricionts$rest2` to decrease.
  - `rest.horz` is a vector with same length as slot `sign` above and specifies the length of periods the restrictions are imposed. If `rest.horz` is 1, only impact restrictions are considered.
  - `constr` is a vector with same length as slot `sign` above with elements lying between `0` and `1`. It specifies the percentage of countries for which cross-country restrictions have to hold. If no cross-country restrictions are supplied, set all elements of `constr` to 1.
  - `scal` optional numeric in case impact normalization is desired.

  #' `MaxTries` Optional numeric corresponding to the maximum tries to search for a rotation matrix that fulfills the user-specified restrictions. Default is set to 7500. After `MaxTries` unsuccessful tries the algorithm sets the impulse response for that specific posterior draw to `NA`. `shock2` define a second list with the same arguments as `shock1` to identify a second shock. Can be used iteratively to identify multiple shocks.

`save.store` If set to `TRUE` the full posterior is returned. Default is set to `FALSE` in order to save storage.

`multithread` If set to `TRUE` parallel computing using the packages `foreach` and `doParallel`. Number of cores is set to maximum number of cores in the computer. This option is recommended when working with sign restrictions to speed up computations. Default is set to `FALSE` and thus no parallelization.

`verbose` If set to `FALSE` it suppresses printing messages to the console.

Value

Returns a list of class `bgvar.irf` with the following elements:

- `posterior` is a four-dimensional array (K times nhor times nr. of shocks times 7) that contains 7 quantiles of the posterior distribution of the impulse response functions: the 50% ("low25" and "high75"), the 68% ("low16" and "high84") and the 90% ("low05" and "high95") credible sets along with the posterior median ("median").
- `rot.nr` in case identification is based on sign restrictions (i.e., `ident="sign"`), this provides the number of rotation matrices found for the number of posterior draws (save*save_thin).
- `shock` in case identification is based on Cholesky decomposition (i.e. `ident="chol"`), this gives back the details of the identification specification.
- `sign.constr` in case identification is based on sign restrictions (i.e. `ident="sign"`), this gives back the set of sign restrictions specified by the user.
- `ident` character giving back the chosen identification scheme.
- `struc.obj` is a list object that contains posterior quantitites needed when calculating historical decomposition and structural errors via `hd.decomp`.
  - `A` median posterior of global coefficient matrix.
  - `Ginv` median posterior of matrix `Ginv`, which describes contemporaneous relationships between countries.
  - `S` posterior median of matrix with country variance-covariance matrices on the main diagonal.
  - `Rmed` posterior rotation matrix if `ident="sign"`.
- `model.obj` is a list object that contains model-specific information, in particular
  - `xglobal` used data of the model.
  - `plag` used lag specification of the model.
- `IRF_store` is a four-dimensional array (K times nhor times nr. of shock times saves) which stores the whole posterior distribution. Exists only if `save.irf.store=TRUE`.

Author(s)

Maximilian Boeck, Martin Feldkircher, Florian Huber

References

Arias, J.E., Rubio-Ramirez, J.F, and D.F. Waggoner (2018) *Inference Based on SVARs Identified with Sign and Zero Restrictions: Theory and Applications.* Econometrica Vol. 86(2), pp. 685-720.

D'Amico, S. and T. B. King (2017) *What Does Anticipated Monetary Policy Do?* Federal Reserve Bank of Chicago Working paper series, Nr. 2015-10.

Pesaran, H.M. and Y. Shin (1998) *Generalized impulse response analysis in linear multivariate models.* Economics Letters, Volume 58, Issue 1, p. 17-29.

Examples

```
oldpar <- par(no.readonly =TRUE)
# First example, a US monetary policy shock, quarterly data
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,prior="SSVS",
                      eigen=TRUE)
# US monetary policy shock
shocks<-list();shocks$var="stir";shocks$cN<-"US";shocks$ident="chol";shocks$scal=-100
irf.chol.us.mp<-IRF(obj=model.ssvs.eer,shock=shocks,nhor=24)
# plots an impulse response function
plot(irf.chol.us.mp,resp="US")

# calculates generalized impulse response functions for the same shock as above
shocks$ident="girf"
irf.girf.ssvs<-IRF(obj=model.ssvs.eer,shock=shocks,nhor=24)
plot(irf.girf.ssvs,resp="US.y")
# Shock to first ordered variable yields same responses of Cholesky and GIRF
shocks<-list();shocks$var="y";shocks$cN<-"US";shocks$ident="chol";shocks$scal<--100
irf.chol<-IRF(model.ssvs.eer,shock=shocks,nhor=24)
shocks$ident<-"girf"
irf.girf<-IRF(model.ssvs.eer,shock=shocks,nhor=24)
matplot(cbind(irf.chol$posterior["US.y",,1,"median"],
              irf.girf$posterior["US.y",,1,"median"]),
        type="l",ylab="")
matplot(cbind(irf.chol$posterior["US.Dp",,1,"median"],
              irf.girf$posterior["US.Dp",,1,"median"]),
        type="l",ylab="")
matplot(cbind(irf.chol$posterior["EA.y",,1,"median"],
              irf.girf$posterior["EA.y",,1,"median"]),
        type="l",ylab="")

sign.constr<-list()
# the variable to shock, can be imposed for more than one country
sign.constr$shock1$shock<-c("US.stir")
# but should be the same variable for all of them
sign.constr$shock1$restrictions$res1<-c("US.y")
sign.constr$shock1$restrictions$res2<-c("US.Dp")
# first entry is for the shock, following entries for the restrictions
# (ltir should go up, y and p go down)
sign.constr$shock1$sign<-c(">","<","<")
# nr. of time periods restrictions are imposed, first entry is for the shock,
# following entries for the restrictions
sign.constr$shock1$rest.horz<-c(1,1,1)
# are constraints binding for all (1) countries specified for
# at least 50\% of the countries (0.5), or 75\% (0.75)
sign.constr$shock1$constr<-c(1,1,1)
# a minus 100 bp shock to long-term interest rates (on average)
sign.constr$shock1$scal=+100
sign.constr$MaxTries<-200
```

```
irf.sign.us.mp<-IRF(obj=model.ssvs.eer,sign.constr=sign.constr,nhor=24)
plot(irf.sign.us.mp,resp=c("US"))

# second example, cross-country restrictions, multiple shocks and ECB country model
data(monthlyData);monthlyData$OC<-NULL
OE.weights <- list(EB=EA.weights)
model.ssvs<-bgvar(Data=monthlyData,W=W,saves=100,burns=100,plag=1,prior="SSVS",
                  thin=1,eigen=TRUE,OE.weights=OE.weights)
EA_countries <- c("AT", "BE", "DE","ES", "FI","FR", "IE", "IT", "NL", "PT","GR","SK")
# A simultaneous Cholesky shock to long-term interest rates in the euro area countries,
# scaled to amount to -100 basis points (on average over the EA countries).
# Note that the ordering of the variables influences the response, the ordering is exactly as
# in the country models, to use a different order you have re-estimate the model (by bgvar)
shocks<-list();shocks$var="ltir";shocks$cN<-EA_countries;shocks$ident="chol";shocks$scal=-100
irf.chol.ssvs<-IRF(obj=model.ssvs,shock=shocks,nhor=48)
plot(irf.chol.ssvs,resp=c("AT"))
# imposes sign restrictions on the cross-section and for a global shock (long-term interest rates)
sign.constr<-list()
# the variable to shock, can be imposed for more than one country
sign.constr$shock1$shock<-c(paste(EA_countries[-c(3,12)],".ltir",sep=""))
# but should be the same variable for all of them
# restrictions (industrial production should decrease for selected countries)
sign.constr$shock1$restrictions$res1<-paste(EA_countries,".y",sep="")
# another set of restrictions (inflation  should decrease for selected countries)
sign.constr$shock1$restrictions$res2<-paste(EA_countries,".p",sep="")
# first entry is for the shock, following entries for the restrictions
# (ltir should go up, y and p go down)
sign.constr$shock1$sign<-c(">","<","<")
# nr. of time periods restrictions are imposed, first entry is for the shock,
# following entries for the restrictions
sign.constr$shock1$rest.horz<-c(1,1,1)
# are constraints binding for all (1) countries specified or for
# at least 50\% of the countries (0.5), or 75\% (0.75)
sign.constr$shock1$constr<-c(1,0.5,0.5)
# a minus 100 bp shock to long-term interest rates (on average)
sign.constr$shock1$scal=-100
sign.constr$MaxTries<-200
irf.sign.ssvs<-IRF(obj=model.ssvs,nhor=48,sign.constr=sign.constr)
plot(irf.sign.ssvs,resp=c("AT"))

# Same example but using a local (German) shock and cross-country restrictions.
# Note that the ordering of the variables influences the response,
# the ordering is exactly as in the country models, to use a different order you have re-estimate
# the model (by bgvar)
shocks<-list();shocks$var="ltir";shocks$cN<-EA_countries;shocks$ident="chol";shocks$scal=-100
irf.chol.ssvs<-IRF(obj=model.ssvs,shock=shocks,nhor=24)

# imposes sign restrictions on the cross-section and for a global shock (long-term interest rates)
sign.constr<-list()
sign.constr$shock1$shock<-c("DE.ltir") # the variable to shock,
# can be imposed for more than one country
#but should be the same variable for all of them
# restrictions (industrial production should decrease for selected countries)
```

```
sign.constr$shock1$restrictions$res1<-paste(EA_countries,".y",sep="")
# another set of restrictions (inflation  should decrease for selected countries)
sign.constr$shock1$restrictions$res2<-paste(EA_countries,".p",sep="")
# first entry is for the shock, following entries for the restrictions
# (ltir should go up, y and p go down)
sign.constr$shock1$sign<-c(">","<","<")
# nr. of time periods restrictions are imposed,
# first entry is for the shock, following entries for the restrictions
sign.constr$shock1$rest.horz<-c(2,2,1)
# are constraints binding for all (1) countries specified or for
# at least 50\% of the countries (0.5), or 75\% (0.75)
sign.constr$shock1$constr<-c(1,0.5,0.5)
# a minus 100 bp shock to long-term interest rates (on average)
sign.constr$shock1$scal=-100
sign.constr$MaxTries<-200
irf.sign.ssvs<-IRF(obj=model.ssvs,nhor=24,sign.constr=sign.constr)

# Example with zero restriction (Arias et al., 2018) and
# rationality conditions (D'Amico and King, 2017).
data("eerDataspf")
model.ssvs.eer<-bgvar(Data=eerDataspf,W=W.trade0012.spf,saves=300,burns=300,
                      plag=1,prior="SSVS",thin=1,eigen=TRUE)
sign.constr<-list()
sign.constr$shock1$shock<-"US.stir_t+4"
sign.constr$shock1$restrictions$res1<-"US.Dp_t+4"
sign.constr$shock1$restrictions$res2<-"US.stir"
sign.constr$shock1$restrictions$res3<-"US.y_t+4"
# rationality condition: US.stir_t+4 on impact is equal to average of
# IRF of US.stir between horizon 1 and 4 (defined with rest.horz, but as period 5!)
sign.constr$shock1$restrictions$res4<-"US.stir_t+4"
# rationality condition: US.Dp_t+4 on impact is equal to H-step ahead IRF of US.Dp in
# horizon 4 (defined with rest.horz, but as period 5!)
sign.constr$shock1$restrictions$res5<-"US.Dp_t+4"
# rationality condition: US.y_t+4 on impact is equal to H-step ahead IRF of US.y in
# horizon 4 (defined with rest.horz, but as period 5!)
sign.constr$shock1$restrictions$res6<-"US.y_t+4"
sign.constr$shock1$sign<-c(">","<","0","<","ratio.avg","ratio.H","ratio.H")
sign.constr$shock1$rest.horz<-c(1,1,1,1,5,5,5)
sign.constr$shock1$constr<-c(1,1,1,1,1,1,1)
sign.constr$shock1$scal=0.1
sign.constr$MaxTries<-100
irf.sign<-IRF(obj=model.ssvs.eer,nhor=20,sign.constr=sign.constr)
par(mfrow=c(4,1),mar=c(5.1,4.1,4.1,2.1))
# rationality condition: US.stir_t+4 on impact is equal to average of IRF of
# US.stir between horizon 1 and 4
matplot(cbind(irf.sign$posterior["US.stir_t+4",,1,"median"],
              irf.sign$posterior["US.stir",,1,"median"]),
        type="l",ylab="",main="stir")
abline(h=mean(irf.sign$posterior["US.stir",2:5,1,"median"]))
abline(v=c(1,5),lty=3,col="grey")
# rationality condition: US.y_t+4 on impact is equal to H-step ahead IRF of US.y in horizon 4
matplot(cbind(irf.sign$posterior["US.y_t+4",,1,"median"],
              irf.sign$posterior["US.y",,1,"median"]),
```

```
        type="l",ylab="",main="y")
abline(h=irf.sign$posterior["US.y_t+4",1,1,"median"])
abline(v=5,lty=3,col="grey")
# rationality condition: US.Dp_t+4 on impact is equal to H-step ahead IRF of US.Dp in horizon 4
matplot(cbind(irf.sign$posterior["US.Dp_t+4",,1,"median"],
              irf.sign$posterior["US.Dp",,1,"median"]),
        type="l",ylab="",main="Dp")
abline(h=irf.sign$posterior["US.Dp_t+4",1,1,"median"])
abline(v=5,lty=3,col="grey")
par(mar=rep(0,4))
plot("1",type="n",axes=FALSE)
legend("center",c("expectation","actual"),lty=1:2,col=c("black","red"),bty="n",ncol=2)
par(oldpar)
```

IRF.cf *Counterfactual Analysis*

Description

Function to perform counterfactual analysis. It enables to neutralize the response of a specific variable to a given shock.

Usage

```
IRF.cf(obj,shockvar,resp,nhor=24,save.store=FALSE,verbose=TRUE)
```

Arguments

`obj`	an object of class `bgvar`.
`shockvar`	structural shock of interest.
`resp`	response variable to neutralize.
`nhor`	forecasting horizon.
`save.store`	If set to `TRUE` the full posterior is returned. Default is set to `FALSE` in order to save storage.
`verbose`	If set to `FALSE` it suppresses printing messages to the console.

Value

Returns a list of class `bgvar.irf` with the following elements:

- `posterior` is a four-dimensional array (K times K times nhor times 7) that contains 7 quantiles of the posterior distribution of the impulse response functions: the 50% ("low25" and "high75"), the 68% ("low16" and "high84") and the 90% ("low05" and "high95") credible sets along with the posterior median ("median").
- `struc.obj` is a list object that contains posterior quantitites needed when calculating historical decomposition and structural errors via `hd.decomp`.

  - A median posterior of global coefficient matrix.
  - Ginv median posterior of matrix Ginv, which describes contemporaneous relationships between countries.
  - S posterior median of matrix with country variance-covariance matrices on the main diagonal.
- model.obj is a list object that contains model-specific information, in particular
  - xglobal used data of the model.
  - plag used lag specification of the model.
- IRF_store is a four-dimensional array (K times nhor times nr. of shock times saves) which stores the whole posterior distribution. Exists only if save.irf.store=TRUE.

Author(s)

Maximilian Boeck, Martin Feldkircher

Examples

```
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,prior="SSVS",
                      eigen=TRUE)

# very time-consuming
## Not run:
irf.cf <- IRF.cf(model.ssvs.eer,shockvar="US.stir",resp="US.rer",nhor=24)

## End(Not run)
```

---

list_to_matrix *Convert Input List to Matrix*

Description

Converts a list to an appropriate input matrix for use of bgvar.

Usage

```
list_to_matrix(datalist)
```

Arguments

datalist	a list of length N which contains each a matrix of size T times k, where T are time periods and k variables per entity..

Details

Note the naming convention. Columns should indicate entity and variable name, separated by a dot, e.g. `US.y`.

Value

returns a matrix of size T times K (number of time periods times number of total variables).

Author(s)

Maximilian Boeck

See Also

bgvar

`logLik.bgvar` *Extract Log-Likelihood*

Description

Extract Log-Likelihood for `bgvar`.

Usage

```
## S3 method for class 'bgvar'
logLik(object, ..., quantile = 0.5)
```

Arguments

`object`	an object of class `bgvar`.
`...`	additional arguments.
`quantile`	reported quantiles. Default is set to median.

Value

Returns an vector of dimension `q` (number of specified quantiles) of global log-likelihoods.

Examples

```
library(BGVAR)
data(eerData)
model.ng <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=100,burns=100)
logLik(model.ng)
```

`lps.bgvar.pred` *Compute log-predictive scores*

Description

Computes and prints log-predictive score of an object of class `bgvar.predict`.

Usage

```
lps(object, ...)
```

Arguments

`object`	an object of class `bgvar.predict`.
`...`	additional arguments.

Value

Returns an object of class `bgvar.lps`, which is a matrix of dimension h times K, whereas h is the forecasting horizon and K is the number of variables in the system.

Author(s)

Maximilian Boeck, Martin Feldkircher

Examples

```
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,prior="SSVS",
                      eigen=TRUE,h=8)
fcast <- predict(model.ssvs.eer,fhorz=8,save.store=TRUE)
lps   <- lps(fcast)
```

`matrix_to_list` *Convert Input Matrix to List*

Description

Converts a big input matrix to an appropriate list for use of `bgvar`.

Usage

```
matrix_to_list(datamat)
```

Arguments

datamat — a matrix of size T times K, where T are time periods and K total amount of variables.

Details

Note the naming convention. Columns should indicate entity and variable name, separated by a dot, e.g. `US.y`.

Value

returns a list of length `N` (number of entities).

Author(s)

Maximilian Boeck

See Also

`bgvar`

---

monthlyData — *Monthly EU / G8 countries macroeconomic dataset*

Description

This data set contains monthly observations on industrial production, consumer price indices, short- and long-term interest rates, real effective exchange rates and equity prices. The time period covered is from January 2000 to December 2015 and the country coverage amounts to 28 countries – roughly corresponding to EU member states + G-8 countries and a country model to model common monetary policy in the euro area.

Usage

```
monthlyData
```

Format

The data loads three objects `monthlyData`, which is a list object of length `N+1` (i.e, the number of countries + the ECB country model), `W`, which is an `N` times `N` weight matrix with rowsums summing up to unity and zero elements on its diagonal. The countries are abbreviated using ISO-2 codes. The weight matrix corresponds to average annual input output flows for the `N` countries over the period from 2000 to 2014. The data are from the world input output table database (`http://www.wiod.org/home`) and are fully described in Timmerman et al. (2015). `monthlyData` contains the country data. Per default, variables that should affect all countries (global variables) are treated as endogenous variables in the US country model (`poil`, `pcom`, `vix`). Akin to Georgiadis (2015), interest setting in the euro area is modeled by a Taylor rule that includes ppp-weighted output and prices of euro area countries. The euro area interest rate enters other country models as an additional exogenous variable. For more details, see below:

- W. N times N weight matrix, rowsums equal unity.
- monthlyData is a list object of length N containing
  - y Industrial production index, in real terms, logarithmic transform and seasonally adjusted.
  - p Harmonized Consumer Price Index (HCPI) for EU member states, for other countries Consumer Price Index. Data in logarithmic transform and seasonally adjusted.
  - stir Short-term interest rate, typically 3 months money market rate.
  - EAstir Short-term interest rate, typically 3 months money market rate (3 months euribor).
  - ltir Long term interest rates, typically 10-year government bond yields.
  - er Real effective exchange rate index, deflated by consumer prices.
  - eq Equity price index, in logarithmic transform.
  - poil Price of oil, seasonally adjusted, in logarithms.
  - pcom Commodity price index, seasonally adjusted, in logarithms.
  - vix Volatility index, in logarithms.

---

pesaranData *pesaranData*

Description

This data set contains 151 quarterly observations by country, spanning the period from 1979Q2 to 2016Q4. It can be downloaded from https://sites.google.com/site/gvarmodelling/gvar-toolbox. The country coverage is 33 countries.

Usage

```
pesaranData
```

Format

The data loads three objects pesaranData, which is a list object of length N (i.e, the number of countries) and W.1316, which is an N times N weight matrix with rowsums summing up to unity and zero elements on its diagonal. The global variable, oil prices, is included in the US country model as e.g., in Dees et al. (2007). The countries are abbreviated using ISO-2 codes. The weight matrix corresponds to average annual bilateral trade flows over the period from 2013 to 2016.peseranData contains the country data in logarithms, for more details, see below:

W.1316 N times N weight matrix based on trade flows, rowsums equal unity.

tA N times N times T array that contains the yearly, bilateral trade flows, which were used to construct W.1316.

peseranData is a list object of length N containing

- y Real GDP.
- Dp Consumer price inflation.

- `eq` Equity prices.
- `stir` Short-term interest rate, typically 3-months money market rate.
- `ltir` Long-term interest rates, typically 10-year government bond yields.
- `poil` Price of oil.
- `pmetal` Price of metals.
- `pmat` Price of agricultural products.

`plot.bgvar` *Plotting function for fitted values*

Description

Plots the fitted values in red of either the country VARs or the GVAR (default) along with the original data.

Usage

```
## S3 method for class 'bgvar'
plot(x, ..., global = TRUE, resp = NULL)
```

Arguments

`x`	an object of class `bgvar`.
`...`	additional arguments.
`global`	if `TRUE` global fitted values are plotted, otherwise country fitted values.
`resp`	if only a subset of variables or countries should be plotted. If set to default value `NULL` all countries/variables are plotted.

Value

No return value.

Examples

```
library(BGVAR)
data(eerData)
model.ssvs <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=100,burns=100,
                    prior="SSVS")
summary(model.ssvs)
plot(model.ssvs, resp="EA")
```

`plot.bgvar.fevd` *Plotting Function for Forecast Error Variance Decomposition*

Description

Plots the decomposition of a specific time series into selected structural shocks.

Usage

```
## S3 method for class 'bgvar.fevd'
plot(x, ..., ts, k.max = 10)
```

Arguments

`x`	an object of class `bgvar.fevd`.
`...`	additional arguments.
`ts`	specify the decomposed time series to be plotted.
`k.max`	plots the k series with the highest for the decomposition of `ts`.

Value

No return value.

Author(s)

Maximilian Boeck

See Also

`IRF`

Examples

```
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,
                      prior="SSVS",thin=1,eigen=TRUE)

# US monetary policy shock
shocks<-list();shocks$var="stir";shocks$cN<-"US";shocks$ident="chol";shocks$scal=-100
irf.chol.us.mp<-IRF(obj=model.ssvs.eer,shock=shocks,nhor=48)

# calculates FEVD for variables US.Dp and EA.y
fevd.us.mp=fevd.decomp(obj=irf.chol.us.mp,var.slct=c("US.Dp","EA.y"))

plot(fevd.us.mp, ts="US.Dp", k.max=10)
```

plot.bgvar.irf *Plot predictions of bgvar*

Description

Plots the predictions of an object of class `bgvar.predict`.

Usage

```
## S3 method for class 'bgvar.irf'
plot(x, ..., resp, shock.nr = 1, cumulative = FALSE)
```

Arguments

`x`	an object of class `bgvar.irf`.
`...`	additional arguments.
`resp`	specify a variable to plot predictions.
`shock.nr`	specify shock to be plotted.
`cumulative`	whether cumulative impulse response functions should be plotted. Default is set to `FALSE`.

Value

No return value.

Author(s)

Maximilian Boeck, Martin Feldkircher

See Also

`IRF`

Examples

```
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,prior="SSVS",
                      eigen=TRUE)
# US monetary policy shock
shocks<-list();shocks$var="stir";shocks$cN<-"US";shocks$ident="chol";shocks$scal=-100
irf.chol.us.mp<-IRF(obj=model.ssvs.eer,shock=shocks,nhor=24)
# plots an impulse response function
plot(irf.chol.us.mp,resp="US.y")
```

plot.bgvar.pred *Plot predictions of bgvar*

Description

Plots the predictions of an object of class `bgvar.predict`.

Usage

```
## S3 method for class 'bgvar.pred'
plot(x, ..., resp = NULL, Cut = 40)
```

Arguments

`x`	an object of class `bgvar.predict`.
`...`	additional arguments.
`resp`	specify a variable to plot predictions.
`Cut`	length of series to be plotted before prediction begins.

Value

No return value.

Author(s)

Maximilian Boeck, Martin Feldkircher

Examples

```
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,prior="SSVS",
                      eigen=TRUE)
fcast <- predict(model.ssvs.eer,fhorz=8,save.store=TRUE)
plot(fcast, resp="US.Dp", Cut=20)
```

plot.bgvar.resid *Plotting function for residuals*

Description

Either plots country-residuals or the global-residuals.

Usage

```
## S3 method for class 'bgvar.resid'
plot(x, ..., global = TRUE, resp = NULL)
```

Arguments

x	an object of class bgvar.res.
...	additional arguments.
global	if TRUE global residuals are plotted, otherwise country residuals.
resp	default to NULL. Either specify a single country or a group of variables to be plotted.

Value

No return value.

Examples

```
library(BGVAR)
data(eerData)
model.ssvs <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=100,burns=100,
                    prior="SSVS")
summary(model.ssvs)
res <- residuals(model.ssvs)
plot(res, resp="EA")
```

predict.bgvar *Compute predictions*

Description

A function that computes predictions based on a object of class bgvar.

Usage

```
## S3 method for class 'bgvar'
predict(object, ..., fhorz = 8, save.store = FALSE, verbose = TRUE)
```

Arguments

object	an object of class bgvar.
...	additional arguments.
fhorz	the forecast horizon.
save.store	If set to TRUE the full distribution is returned. Default is set to FALSE in order to save storage.
verbose	If set to FALSE it suppresses printing messages to the console.

Value

Returns an object of class bgvar.pred with the following elements

- fcast is a K times fhorz times 5-dimensional array that contains 16%th, 25%th, 50%th, 75%th and 84% percentiles of the posterior predictive distribution.
- xglobal is a matrix object of dimension T times N (T # of observations, K # of variables in the system).
- fhorz specified forecast horizon.
- lps.stats is an array object of dimension K times 2 times fhorz and contains the mean and standard deviation of the log-predictive scores for each variable and each forecast horizon.
- hold.out if h is not set to zero, this contains the hold-out sample.

Author(s)

Maximilian Boeck, Martin Feldkircher, Florian Huber

Examples

```
library(BGVAR)
data(eerData)
model.ssvs <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=100,burns=100,
                    prior="SSVS")
fcast <- predict(model.ssvs, fhorz=8)
```

---

print.bgvar *Print bgvar Output*

Description

print prints the main arguments of an bgvar object.

Usage

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

Arguments

x	an object of class bgvar.
...	other arguments.

Value

No return value.

Author(s)

Maximilian Boeck, Martin Feldkircher

See Also

bgvar to estimate a bgvar object.

Examples

```
library(BGVAR)
data(eerData)
model.ssvs <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=100,burns=100,
                    prior="SSVS",SV=FALSE,hyperpara=hyperpara,thin=1)
print(model.ssvs)
```

---

print.bgvar.CD *Print convergence diagnostics*

Description

Print convergence diagnostics

Usage

```
## S3 method for class 'bgvar.CD'
print(x, ...)
```

Arguments

x	object of class bgvar.CD.
...	additional arguments.

Value

No return value.

print.bgvar.lps *Print prediction evaulation*

Description

print prints log-predictive scores (LPS) of out-of-sample predictions computed with bgvar.predict.

Usage

```
## S3 method for class 'bgvar.lps'
print(x, ...)
```

Arguments

x	an object of class bgvar.lps.
...	other arguments.

Value

No return value.

Author(s)

Maximilian Boeck, Martin Feldkircher

See Also

bgvar to estimate a bgvar object and predict.bgvar to compute predictions.

Examples

```
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,prior="SSVS",
                      eigen=TRUE,h=8)
fcast <- predict(model.ssvs.eer,fhorz=8,save.store=TRUE)
lps(fcast)
```

print.bgvar.rmse *Print prediction evaulation*

Description

`print` prints root mean squared errors (RMSE) of out-of-sample predictions computed with `bgvar.predict`.

Usage

```
## S3 method for class 'bgvar.rmse'
print(x, ...)
```

Arguments

`x`	an object of class `bgvar.predeval`.
`...`	other arguments.

Value

No return value.

Author(s)

Maximilian Boeck, Martin Feldkircher

See Also

`bgvar` to estimate a `bgvar` object and `predict.bgvar` to compute predictions.

Examples

```
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,prior="SSVS",
                      eigen=TRUE,h=8)
fcast <- predict(model.ssvs.eer,fhorz=8,save.store=TRUE)
rmse(fcast)
```

`resid.corr.test` *Residual autocorrelation test*

Description

An F-test for serial autocorrelation in the residuals.

Usage

```
residual.corr.test(obj, lag.cor=1, alpha=0.95, dig1=5, dig2=3)
```

Arguments

`obj`	an object of class `bgvar`.
`lag.cor`	the order of serial correlation to be tested for. Default is set to `lag.cor=1`.
`alpha`	significance level of test. Default is set to `alpha=0.95`.
`dig1`	number of digits to display F-statistics and its critical values.
`dig2`	number of digits to display p-values.

Details

It is the F-test of the familiar Lagrange Multiplier (LM) statistic (see Godfrey 1978a, 1978b), also known as the 'modified LM' statistic. The null hypothesis is that rho, the autoregressive parameter on the residuals, equals 0 indicating absence of serial autocorrelation. For higher order serial correlation, the null is that all rho's jointly are 0. The test is implemented as in Vanessa Smith's and Alessandra Galesi's "GVAR toolbox 2.0 User Guide", page 129.

Value

Returns a list with the following objects

- `Fstat` contains a list of length `N` with the associated F-statistic for each variable in each country.
- `resTest` contains a matrix of size 2N times K+3, with the F-statistics for each country and each variable.
- `p.res` contains a table which summarizes the output.
- `pL` contains a list of length `N` with the associated p-values for each variable in each country.

Author(s)

Martin Feldkircher

References

Godfrey, L.G. (1978a) *Testing Against General Autoregressive and Moving Average Error Models When the Regressors Include Lagged Dependent Variables.* Econometrica, 46, pp. 1293-1302. Godfrey, L.G. (1978b) *Testing for Higher Order Serial Correlation in Regression Equations When the Regressors Include Lagged Dependent Variables.* Econometrica, 46, pp. 1303-1310. Smith, L. V. and A. Galesi (2014) *GVAR Toolbox 2.0 User Guide*, available at https://sites.google.com/site/gvarmodelling/gvar-toolbox.

See Also

print.bgvar

Examples

```
library(BGVAR)
data(eerData)
model.mn <- bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,prior="MN")
residual.corr.test(model.mn)
```

---

residuals.bgvar *Extract residuals of Global Vector Autoregression*

Description

Calculate residuals of the global model and the country models.

Usage

```
## S3 method for class 'bgvar'
residuals(object, ...)

## S3 method for class 'bgvar'
resid(object, ...)
```

Arguments

object	a fitted bgvar object.
...	other arguments.

Details

This function calculates residuals of the global and the country models based on a bgvar object. Country models' residuals are equivalent to output generated by the print.bgvar function in case no trimming has been used. If trimming was invoked to discard unstable draws output of both functions might differ since print.bgvar calculates residuals as a running mean to save storage which is based on the *whole* set of posterior draws (including discarded draws). In this case it is recommended to recalculate the residuals with `residuals.bgvar` and re-do the serial autocorrelation or average pairwise cross-correlation analysis using functions `resid.corr.test` and `avg.pair.cc`.

Value

returns a list with the following arguments

- `global` is a (T-p) times K times saves/thin array containing the residuals of the global model.
- `country` is a (T-p) times K times saves/thin array containing the residuals of the country models.
- `Data` is a (T-p) times K matrix containing the data of the model.

Author(s)

Maximilian Boeck, Martin Feldkircher

See Also

`bgvar` for estimation of a `bgvar` object.

Examples

```
library(BGVAR)
data(eerData)
model.ssvs <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=100,burns=100,
                    prior="SSVS")
res <- residuals(model.ssvs)


resid(model.ssvs)
```

`rmse.bgvar.predict` *Compute root mean squared errors*

Description

Computes and prints root mean squared errors (RMSEs) of an object of class `bgvar.predict`.

Usage

```
rmse(object, ...)
```

Arguments

`object`	an object of class `bgvar.predict`.
`...`	additional arguments.

Value

Returns an object of class `bgvar.rmse`, which is a matrix of dimension h times K, whereas h is the forecasting horizon and K is the number of variables in the system.

Author(s)

Maximilian Boeck, Martin Feldkircher

Examples

```
library(BGVAR)
data(eerData)
model.ssvs.eer<-bgvar(Data=eerData,W=W.trade0012,saves=100,burns=100,plag=1,prior="SSVS",
                      eigen=TRUE,h=8)
fcast <- predict(model.ssvs.eer,fhorz=8,save.store=TRUE)
rmse   <- rmse(fcast)
```

`summary.bgvar` *Summarizing Bayesian Global Vector Autoregression Fits*

Description

Output gives model information as well as some descriptive statistics on convergence properties, likelihood, serial autocorrelation in the errors and the average pairwise autocorrelation of cross-country residuals.

Usage

```
## S3 method for class 'bgvar'
summary(object, ...)
```

Arguments

`object`	an object of class `bgvar`.
`...`	other arguments.

Value

No return value.

Author(s)

Maximilian Boeck

See Also

`bgvar` to estimate a `bgvar` object.

`avg.pair.cc` to compute average pairwise cross-country correlation of cross-country residuals separately.

`resid.corr.test` to compute F-test on first-order autocorrelation of cross-country residuals separately.

Examples

```
set.seed(571)
library(BGVAR)
data(eerData)
model.ssvs <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=100,burns=100,
                    prior="SSVS",thin=1,SV=TRUE,trend=TRUE)
summary(model.ssvs)
```

---

vcov.bgvar — *Extract variance-covariance matrix*

Description

Extracts the global variance-covariance matrix for `bgvar` for certain quantiles of the posterior distribution.

Usage

```
## S3 method for class 'bgvar'
vcov(object, ..., quantile = 0.5)
```

Arguments

`object`	an object of class `bgvar`.
`...`	additional arguments.
`quantile`	reported quantiles. Default is set to median.

Value

Returns an `q` times `K` times `K` array of the global variance-covariance matrix, where `q` is the number of specified quantiles (this dimension is dropped if `q=1`) and `K` the number of endogenous variables.

Examples

```
library(BGVAR)
data(eerData)
model.ng <- bgvar(Data=eerData,W=W.trade0012,plag=1,saves=100,burns=100)
vcov(model.ng)
```

Index

∗Topic **datasets**
eerData, 17
eerDataspf, 18
monthlyData, 34
pesaranData, 35

AIC.bgvar, 3
avg.pair.cc, 4, *49*

bgvar, *5*, 5, *22*, *24*, *32*, *34*, *42–44*, *47*, *49*
BIC.bgvar, 12

coef.bgvar, 13
coefficients.bgvar (coef.bgvar), 13
cond.pred, 14
cond.predict (cond.pred), 14
conv.diag, 15

DIC, 16
doParallel, *9*, *22*, *25*

EA.weights (monthlyData), 34
eerData, 17
eerDataspf, 18

fevd.decomp, 19
fitted.bgvar, 21
foreach, *9*, *22*, *25*

geweke.diag, *16*
gfevd.decomp, 21

hd.decomp, 23

IRF, *20*, *24*, 24, *37*, *38*
IRF.cf, 30

list_to_matrix, 31
logLik.bgvar, 32
lps (lps.bgvar.pred), 33
lps.bgvar.pred, 33

matrix_to_list, 33
monthlyData, 34

OC.weights (monthlyData), 34

pesaranData, 35
plot.bgvar, 36
plot.bgvar.fevd, 37
plot.bgvar.irf, 38
plot.bgvar.pred, 39
plot.bgvar.resid, 40
predict.bgvar, 40, *43*, *44*
print (print.bgvar), 41
print.bgvar, 41, *46*
print.bgvar.CD, 42
print.bgvar.lps, 43
print.bgvar.rmse, 44

resid.bgvar (residuals.bgvar), 46
resid.corr.test, 45, *49*
residual.corr.test (resid.corr.test), 45
residuals, *5*
residuals (residuals.bgvar), 46
residuals.bgvar, 46
rmse (rmse.bgvar.predict), 47
rmse.bgvar.predict, 47

summary (summary.bgvar), 48
summary.bgvar, 48

tA (pesaranData), 35

vcov.bgvar, 49

W (monthlyData), 34
W.1316 (pesaranData), 35
W.list (eerData), 17
W.trade0012 (eerData), 17
W.trade0012.spf (eerDataspf), 18