Package 'MRCIGVAR'

December 30, 2023

Title Multi Regime Cointegrated Global VAR Modelling

Version 0.0.0.9000

Description This package contains functions for data generation with flexible model specifications, parameter estimation, model selection, impulse response functions, and parameter constraint tests for the following classes of econometric models including VAR, CIVAR, MRVAR, MRCIVAR, GVAR, CIGVAR, MRGVAR, and MRCIGVAR.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Depends R (>= 2.10)

Imports tseries,
vars,
Matrix,
ggplot2,
gridExtra,
stats

Suggests knitr,
devtools,
roxygen2,
testthat,
rmarkdown

LazyData true

VignetteBuilder knitr

URL https://github.com/puchen8229/MRCIGVAR

BugReports https://github.com/puchen8229/MRCIGVAR/issues

R topics documented:

ABC_MRCIVARestm — *Test of restrictions in the cointegration space of a MRCIVAR model*

Description

Test of restrictions in the cointegration space of a MRCIVAR model

Usage

```
ABC_MRCIVARestm(res = res, H = H, h = h, phi = phi, G = G, psi = psi)
```

Arguments

`res`	An MRCIVAR object of the output of MRCIGVARDatam
`H`	A matrix specifying restrictions on beta
`h`	A vector specifying restrictions on beta
`phi`	A vector of freely varying parameters in beta
`G`	A matrix specifying restrictions on alpha
`psi`	A vector of freely varying parameters in alpha

Value

A list containing estimated MRVECM under restrictions, restricted parameter, and likelihood test statistic

ABSVAR — *This function solves AB_SVAR from the reduced form and return the A, B matrices and the sum of squared errors*

Description

This function solves AB_SVAR from the reduced form and return the A, B matrices and the sum of squared errors

Usage

```
ABSVAR(x0, A0, B0, Sigma)
```

Arguments

`x0`	: difference between the reduced form and the AB form
`A0`	: A matrix in an AB_SVAR model
`B0`	: B matrix in an AB_SVAR model
`Sigma`	: Covariance matrix of the reduced form

Value

a list contains A B and the difference

CCIVARData *Conditional cointegrated VAR*

Description

The function generates data from a conditional cointegrated VAR(p)

Usage

```
CCIVARData(n1, n2, crk, p, T, type, Bc = NA)
```

Arguments

n1	dimension of the conditional cointegrated process
n2	dimension of the conditioning variables
crk	the cointegration rank
p	lag
T	number of observations
type	types of the deterministic component. "none" and "const" are two options.
Bc	Coefficient matrix of the joint cointegration process

Value

a CCIVAR object which is a list of (n1,n2,p,type,r_np,By,Bx,Cy,Sigma,Y,X,resid,U,check,crk,Bc,Cc)

Examples

```
T = 100
res_d <- CCIVARData(n1=4,n2=3,crk=3,p=3,T=T,type="const")
res_e <- CCIVARest(res=res_d)
res_e$Summary
```

CCIVARest *Estimation of CCIVAR*

Description

This function estimates parameters of a specified conditional cointegrated VAR based on provided data.

Usage

```
CCIVARest(res)
```

Arguments

res	a CCIVAR object that can be an output of CCIVARData containing at least n1,n2, Y, X, and crk.

Value

a CCIVAR object with estimated parameter values, AIC, BIC, conditional VECM in a regression format.

Examples

```
T = 100
res_d <- CCIVARData(n1=4,n2=3,crk=3,p=3,T=T,type="const")
res_e <- CCIVARest(res=res_d)
res_e$Summary
```

`CIGVARData` *Data generating process of CIGVAR(m,n,p)*

Description

This function generates data from a cointegrated global VAR process and returns an CIGVAR(m,n,p) object that is a list containing data and parameters used in the CIGVAR(m,n,p) process.

Usage

```
CIGVARData(
  m,
  n,
  p,
  T,
  W = NA,
  r_npo = NA,
  Ao = NA,
  Bo = NA,
  Co = NA,
  Uo = NA,
  Sigmao = NA,
  type = NA,
  X = NA,
  mu = NA,
  DFYflag = NA,
  crk = NA,
  Ncommfakt = 1,
  A = NA,
  uz = NA
)
```

Arguments

m	: number of variables in each country/unit
n	: number of countries/units
p	: an (n x 3) matrix, each raw specifies the lag length of the domestic variables, the foreign variables and the number of the exogenous variables.

T	: number of observations.
	(m,n,p,T) are parameters which must be provided.
W	: an (n x n) weighting matrix. w_ij is the weight of foreign country j in the foreign variables of i-th country diag(W)=0
r_npo	: an (m, p, n) array collecting the roots of the characteristic functions in the lag operator of the country VAR. The number of ones in each ith row of r_npo is the number of unit roots i-th country/unit.
Ao	: an (m, m, p, n) array collecting the coefficients of foreign variables.
Bo	: an (m, m, p, n) array collecting the coefficients of domestic variables.
Co	: an (m , k+1, n) array collecting the coefficients of the deterministic components of the n countries.
Uo	: an (T x mn) matrix of the temporally independent innovation processes
Sigmao	: (mn x mn) matrix of the covariance matrix of the CIGVAR(m,n,p) (W,r_npo,Ao,Bo,Uo,Sigmao) if not provided, they will be generated randomly. The default assumption is one unit root in one country. Hence m-1 cointegration relations in each country.
type	: deterministic component "const", "none", "exog0", and "exog1" are four options
X	: a (T x k x n) array of exogenous stationary variables.
mu	: if type = "const" mu has the same dimension as Co. is an muV is nm vector of the means of the time series in the system
DFYflag	: DFYflag = 0 implies foreign variables are not in the cointegration space. DFYflag = 1 allows foreign variables enter the cointegration relation.
crk	: n vector containing the cointegration rank in each country/unit. crk is to specified for estimation of parameters.
Ncommfakt	: number of common exogenous stochastic factors
A	: a transformation matrix that adds the common exogenous factors to the system.
uz	: innovations of the exogenous common factors

Value

a CIGVAR object containing the generated data, the parameters used and the exogenous variables.

Examples

```
n = 5
p = (1:15)*0; dim(p) = c(5,3)
p[,1] = 3; p[,2]=3;
res_d = CIGVARData(m=4,n=5,p=p,T=100,type="const")
max(abs(res_d$Y))
plot(ts(res_d$Y[,1:10]))
res_d$r_npo
STAT(res_d$G)
res_e = CIGVARest(res_d)
res_e$Summary
STAT(res_d$G)
plot(ts(res_d$Y[,1:10]))
```

CIGVARest *Estimation of CIGVAR(res)*

Description

This function estimates the parameters of a CIGVAR(m,n,p) object based on provided data. It runs a VECM estimation country/unit by country/unit under a given cointegration rank and pieces the results together to obtain a CIGVAR.

Usage

```
CIGVARest(res)
```

Arguments

res	a CIGVAR object of an output of CIGVARData including at least values of m,n,p,type,Y,crk and optionally X.

Value

a CIGVAR object with estimated parameter values, AIC, BIC and LH

Examples

```
n = 5
p = (1:15)*0; dim(p) = c(5,3)
p[,1] = 3; p[,2]=3;
r_npo = (1:(3 * 3 * 5))*0; dim(r_npo) = c(3,3,5)
r_npo[,,1] = matrix(c(1,1,3,2,3,3,3,3,3),3,3)
r_npo[,,2] = matrix(c(1,2,3,2,3,3,3,3,3),3,3)
r_npo[,,3] = matrix(c(1,2,3,2,3,3,3,3,3),3,3)
r_npo[,,4] = matrix(c(1,2,3,2,3,3,3,3,3),3,3)
r_npo[,,5] = matrix(c(1,2,3,2,3,3,3,3,3),3,3)

res_d = CIGVARData(m=3,n=5,p=p,T=500,r_npo=r_npo,type="const",DFYflag=0,Ncommfakt=1)
max(res_d$Y)
STAT(res_d$G)
plot(ts(res_d$Y[,1:10]))
res_e = CIGVARest(res_d)
res_e$Summary
res_e$Summary$CRK_Test
## adjust the model specification according to the cointegration rank test
res_d$crk = c(1,2,2,2,2)
res_e = CIGVARest(res_d)
res_e$Summary
## unit roots in the CIGVAR model
STAT(res_e$G)
```

CIGVAR_Select *Calculation of information criteria for CIGVAR models*

Description

This function calculates AIC and BIC criteria for a range of lag specifications of the domestic and foreign variables up to the maximum given in L_V.

Usage

```
CIGVAR_Select(res, L_V = L_V)
```

Arguments

res	: an object generated from CIGVARest.
L_V	: a 2 components vector containing the maxima of the domestic lag and the foreign lag, respectively.

Value

a matrix with different lag specifications and the corresponding information criteria.

Examples

```
n = 5
p = (1:15)*0; dim(p) = c(5,3)
p[,1] = 3; p[,2]=3;
p[2,1] = 2;  p[3,1] = 2
res_d = CIGVARData(m=2,n=5,p=p,T=5000,type="const",DFYflag=0)
res_d$r_npo
res_d$check

plot(ts(res_d$Y[1:100,1:10]))

res_e = CIGVARest(res_d)
res_e$Summary
res_e$crk = c(1,1,1,1,1)
L_V = c(4,4)
CIGVARSelecte = CIGVAR_Select(res=res_e,L_V=c(4,4))
CIGVARSelecte[which.min(CIGVARSelecte[,17]),]
CIGVARSelecte[which.min(CIGVARSelecte[,19]),]
```

CIVARData *Data generating process of CIVAR(p)*

Description

This function generates data from a cointegrated vector autoregressive process CIVAR(p) and returns CIVAR(p) object that is a list containing data and parameters used in the CIVAR(p) process.

Usage

```
CIVARData(n, p, T, r_np, A, B, Co, C1, U, Sigma, type, X, mu, Yo, crk)
```

Arguments

n	: number of variables
p	: lag length
T	: number of observations
r_np	: an (n x p) matrix of roots of the characteristic polynomials of n independent AR(p)-processes. An element 1 in a row implies a unit root process. If the matrix is not provided, it will be generated randomly with one unit root in the first row.
A	: an (n x n) full rank transformation matrix that is used to generate correlated a CIVAR(p) from the n independent AR(p) processes with unit roots.
B	: an (n,n,p) array containing the coefficients of the CIVAR(p) process. If B is not given, it will be calculated out of r_np and A.
Co	: an (n,k+1) matrix containing coefficients of deterministic components in a CIVAR(p) process. For type="none" Co = 0*(1:n), for type="const" Co is an n vector, for type="exog0" Co is a (n,k) matrix, and for type="exog1" Co is an (n,1+k) matrix.
C1	: an (n,1) matrix containing coefficients of the trend component. C1 = 0 if there is no trend in the data.
U	: residuals, if it is not NA it will be used as input to generate the CIVAR(p)
Sigma	: an n x n covariance matrix of the CIVAR(p)
type	: types of deterministic components. "none","rconst","const","rtrend", "trend", "exog0" and "exog1" are 7 options • none: No deterministic component in VECM (Case I) • rconst: An intercept restricted in the cointegration space (Case II) • const: An unrestricted intercept (Case III) • rtrend: An trend component restricted in the cointegration space and an unrestricted intercept in VECM (Case IV) • trend: An unrestricted trend component and an unretricted intercept in VECM (Case V) • exog0: Model with exogeneous variable and no deterministic components • exog1: Model with exogeneous variable and an unrestricted intercept
X	: a (T x k) matrix of exogenous variables.
mu	: an n vector of drifts of the CIVAR(p) process
Yo	: (p x n) matrix of initial values of the process
crk	: number of cointegration relations. It equals n-r, where r is the number of unit roots.

Value

An object of CIVAR(p) containing the generated data, the parameters used and the exogenous variables. res = list(n,p,type,r_np,Phi,A,B,Co,Sigma,Y,X,resid,U,Y1,Yo,check)

Examples

```
res_d = CIVARData(n=2,p=2,T=100,type="const")
res_d = CIVARData(n=2,p=2,T=10,Co=c(1:2)*0,type="none")
res_d = CIVARData(n=2,p=2,T=10,Co=c(1:2)*NA,  type="const")

p = 3
n = 4
r_np = matrix(c(1,2,1.5,1.5,2.5,2.5,2,-1.5,2,-4,1.9,-2.1),4,3)

res_d  = CIVARData(n=4,p=3,T=200,r_np=r_np,Co=matrix(0,n,1) ,type="none",crk=3)
res_e  = CIVARest(res=res_d)
sum(abs(res_e$B-res_d$B))
sum(abs(res_e$Co-res_d$Co))


res_d  = CIVARData(n=4,p=3,T=200)
res_e = CIVARest(res=res_d)
sum(abs(res_e$B-res_d$B))
sum(abs(res_e$Co-res_d$Co))
plot(ts(res_d$Y))
```

CIVARest *Estimation of CIVAR(p)*

Description

This function estimates parameters of a specified CIVAR(p) model based on provided data.

Usage

```
CIVARest(res)
```

Arguments

`res`	:an object of CIVAR(p) containing the components which are the output of CIVARData including at least: n, p, Y, crk, and optionally X and type.

Value

res an object of CIVAR(p) containing estimated parameter values, AIC, BIC, LH and the estimated VECM in regression format.

Examples

```
p = 3
n = 4
r_np = matrix(c(1,2,1.5,1.5,2.5,2.5,2,-1.5,2,-4,1.9,-2.1),4,3)

res_d  = CIVARData(n=4,p=3,T=200,r_np=r_np,Co =(1:n)/(1:n)*0,type="none",crk=3)
res_e  = CIVARest(res=res_d)
res_d  = CIVARData(n=4,p=3,T=200)
B = res_d$B
plot(ts(res_d$Y))
res_d$Co
res_d$type
res_d$crk
res_e = CIVARest(res=res_d)
```

`CIVARTest` *Test of restrictions in the cointegration space and estimation of restricted VECM*

Description

This function estimates constrained VECM using the iteration procedure proposed in Boswijk and Doornik (2003)

Usage

```
CIVARTest(res = res, H = H, h = h, phi = phi, G = G, Dxflag = 0)
```

Arguments

`res`	an estimated CIVAR object with without restrictions
`H`	the restriction matrix on beta
`h`	the restriction vector on beta (free-varying parameters of beta)
`phi`	the restriction vector on beta
`G`	the restriction matrix on alpha
`Dxflag`	A flag that indicates whether X enters the cointegration space.

Value

a list containing (VECMRS, beter, alphae, LSKOEFR, LR and error)

Details

This function runs a likelihood ratio test of linear restrictions on α and β in a CIVAR model in the following form:

$$vec(\alpha') = G\psi, vec(\beta) = H\phi + h$$

example 1 (restrictions on alpha) test of exogeneity: one weakly exogenous variable vec(alpha) is n*crk x 1 vector G is n*crk x (n-k)*crk matrix (k is the number of weakly exogenous) psi (n-k)*crk x 1 vector vec(beta) is n*crk x 1 vector H is n*crk x n*crk identity matrix phi n*crk x 1 vector h n*crk x 1 zero matrix implying ver(beta) = phi. (H is identity and h is zero vector implies only restrictions on alpha)

example 2 (restrictions on beta) test of PPP vec(alpha) is n*crk x 1 vector G is n*crk x n*crk identity matrix, implying no restriction on alpha psi n*crk x 1 vector vec(beta) n*crk x 1 vector H n*crk x 2 matrix that picks out the elements under restrictions two columns out of the identity matrix. ones in a row of H and zero in the corresponding h implies non-restricted beta. phi 2 x 1 vector h n*crk x 1 non zero elements in this vector together with the zero elements in the corresponding row in H are the normalization conditions. (H is identity and h is zero vector implies only restrictions on alpha)

Examples

```
X = matrix(stats::rnorm(2*207),207,2)
colnames(X) = c("ex1","ex2")
res_d  = CIVARData(n=6,p=2,T=207,crk=2,type="exog1", X=as.matrix(X))
colnames(res_d$Y) =  c("w","p","U_l","r","yn","y")
res_e =  CIVARest(res=res_d)
res_e$tst$erg
res_e$tst$beta
res_e$tst$VECMS
n = 6;  crk = 2
G = diag(n*crk); G0 = G; psi = matrix(1,n*crk,1); psi0=psi;   ### No restrictions on alpha
H = diag(n*crk); H2 = H[,-seq(1,n*crk,n)];
h2 = matrix(0,n*crk,1);
h2[seq(1,n*crk,n),1] = rep(1,crk);
phi2 = matrix(1,(n-1)*crk,1)                         ### no restrictions but normalization

G0%*%psi0; H2%*%phi2+h2

ABtest = CIVARTest(res=res_d,H=H2,h=h2,phi=phi2,G=G0,Dxflag=0)
ABtest$betar
ABtest$alphar
ABtest$VECMR$coefficients
ABtest$LR
#1-pchisq(ABtest$LR,2)   ### the fourth is the used restrictions


res_d  = CIVARData(n=7,p=2,T=207,crk=2,type="const")
colnames(res_d$Y) =  c("w","p","U_l","r","yn","y","fsi")
res_e =  CIVARest(res=res_d)
res_e$tst$erg
res_e$tst$beta
res_e$tst$VECMS

n = 7;  crk = 2
G = diag(n*crk); G0 = G; psi = matrix(1,n*crk,1); psi0=psi;   ### No restrictions on alpha

H = diag(n*crk); H2 = H[,-c(1,8)]; h2 = matrix(0,n*crk,1);
h2[c(1,8),1] = c(1,1);  phi2 = matrix(1,12,1)                 ### normalization

G0%*%psi0; H2%*%phi2+h2

ABtest = CIVARTest(res=res_d,H=H2,h=h2,phi=phi2,G=G0,Dxflag=1)
ABtest$LR
ABtest$betar
ABtest$alphar
```

```
#1-pchisq(ABtest$LR,4)
```

`CVECM2CVAR` *Transformation of the estimated parameters of a conditional VECM into the parameters of the corresponding conditional VAR*

Description

Transformation of the estimated parameters of a conditional VECM into the parameters of the corresponding conditional VAR

Usage

```
CVECM2CVAR(param, beta, p = c(1, 2, 2, 2, 2), s = NA, N2)
```

Arguments

`param`	estimated parameters of a conditional VECM
`beta`	the estimated cointegration vecters
`p`	a vector specifying different types of conditional VECM
`s`	indicator variable of different regimes
`N2`	dimension of the conditioning variables

Value

A list of (A, B, C) with B the conditional VAR parameter matrix, C the parameter of the deterministic components, and A the parameter matrix of foreign variables for CIGVAR.

`girf_MRCIGVAR_RM_CB` *Generalized impulse response functions of MRCIGVAR(m,p,n,S,W,TH) with regime migrations*

Description

This function calculates the generalized impulse response functions of an estimated MRVAR(n,p,S) for given a shock vector and initial values.

Usage

```
girf_MRCIGVAR_RM_CB(
  res,
  shock,
  R,
  nstep,
  Omega_hist = NA,
  resid_method = "parametric",
  conf,
  N
)
```

Arguments

res	: a MRCIGVAR object containing the components of an output of MRCIGVARest.
shock	: an mn-vector containing the shocks as impulse.
R	: the number of runs to integrate out the random effects in order to obtain the means (see equation above).
nstep	: the length of the responses
Omega_hist	: the initial values from which the simulation runs of impulse and response functions start
resid_method	: resid_method = c("resid", "parametric"), It generate random residuals either from residuals bootstrap or parametric bootstrap.
conf	: a two component vector containing the tail probabilities of the bootstrap confidence interval.
N	: number of bootstrapping runs

Details

```
GIRF(shock=SHCK) = mean(Y(resid)) - mean(Y(SHCK))
```

It also generates the bootstrapped confidence intervals.

Value

an (n x n x nstep+1 x 3) array containing of impulse response functions with lower and upper confidence bonds. The rows represent response the columns represent impulses.

Examples

```
m = 2
n = 3
p = c(2,2,2,2,2,2,2,2,2,2,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,0,0,0,0,0); dim(p) = c(5,3,2)
p = p[1:n,,]; p[,1,] = 3; p[,2,] = 2

TH = c(1:n)*0; dim(TH) = c(1,n)
SESVI=rep(1,3,5)
r  = rep(1,n)

## case of n = 3, m = 2, S = 2    m: number of variables, n: number of countries
res_d <- MRCIGVARData(m=2,n=3,p=p,TH=TH,T=200,S=2, SESVI=c(1,3,5),r=rep(1,3),Ncommtrend=1)
max(abs(res_d$Y))
STAT(res_d$Go[,,,2])
STAT(res_d$Go[,,,1])
res_e  = MRCIGVARest(res=res_d)

STAT(res_e$Go[,,,2])
STAT(res_e$Go[,,,1])

plot(ts(res_d$Y))
#res_e$Summary

if (!((max(Mod(STAT(res_e$Go[,,,1])))>1)|(max(Mod(STAT(res_e$Go[,,,2])))>1)) ) {
  GIRF <- girf_MRCIGVAR_RM(res=res_e,shock=c(1,1,1,1,1,1),R=100,nstep=10,Omega_hist=NA,
  resid_method="parametric")
```

```
  GIRF_CB <- girf_MRCIGVAR_RM_CB(res=res_e,shock=c(1,1,1,1,1,1),R=100,nstep=10,Omega_hist=NA,
   resid_method="parametric",conf=c(0.05,0.95),N=100)
   IRF_g = IRF_graph(GIRF_CB)
}
```

`girf_MRCIVAR_RM_CB` *Generalized impulse response functions of MRCIVAR(n,p,S) with regime migrations with a confidence interval*

Description

This function calculates the generalized impulse response functions of an estimated MRVAR(n,p,S) with a bootstrapping confidence interval.

Usage

```
girf_MRCIVAR_RM_CB(
  res,
  shock,
  R,
  nstep,
  Omega_hist = NA,
  resid_method = "parametric",
  conf_level,
  N
)
```

Arguments

res	: an MRCIVAR object containing the components of an output of MRCIVARestm1.
shock	: an n vector containing the shocks as impulse.
R	: the number runs to integrate out the random effects in order to obtain the means (see equation above).
nstep	: the length of the responses
Omega_hist	: a (P x n) matrix of initial values, from which the impulse response functions start. Omega_hist determines from which regime the impulse response functions start. For Omega_hist=NA, the impulse response functions will start from the most resent observations.
resid_method	: resid_method = c("resid", "parametric"), It generate the random residuals from residuals bootstrap or parametric bootstrap.
conf_level	: a vecter contain the level of confidences
N	: the number of bootstrap runs in containing the bootstrapped confidence intervals.

Details

For a given shock vector SHCK:

GIRF(shock=SHCK) = mean(Y(resid)) - mean(Y(SHCK))

See H.H. Pesaran and Y. Shin (1998) Generalized impulse response analysis in linear multivariate models, Economics Letters, 58(1) p. 17-29. and G. Koop, M. H. Pesaran, and S. M. Potter (1996), Impulse response analysis in nonlinear multivariate models, Journal of Econometrics, 74 (1996) 119-74.

Value

an (n, n, nstep+1,3) array containing the impulse response functions with lower and upper confidence bonds. The rows represent response, and the columns represent impulses.

Examples

```
n =4

Sigma = 1:(n*n*2)
dim(Sigma) = c(n,n,2)
Sigma[,,1] = diag(n)
Sigma[,,2] = diag(n)
p=matrix(0,2,2)
p[,1] = c(3,3)
res_d = MRCIVARDatam(n=n,p=p,T=250,S=2,SESVI=1,TH=0,Sigmao=Sigma,type="const",r=2)
#colnames(res_d$Y) = c("w","p","I","Q")
max(abs(res_d$Y))
res_e = MRCIVARestm1(res=res_d)
res_e$Summary
Mod(STAT(res_e$B[,,,1]))
Mod(STAT(res_e$B[,,,2]))

#res_e$Summary
if (! max(Mod(STAT(res_e$Bo[,,,1])),Mod(STAT(res_e$Bo[,,,2])) ) > 1.0001 ) {

 IRF  = irf_MRCIVAR_CB(res_e,nstep=20,irf="gen1",runs=10,comb=NA,G=NA,conf=c(0.05,0.95))
 IRF_g1 <- IRF_graph(IRF[[1]])        # irf of regime 1
 IRF_g2<- IRF_graph(IRF[[2]])         # irf of regime 2

}
```

`girf_MRGVAR_RM_CB` *Generalized impulse response functions of MRGVAR(m,p,n,S,W,TH) with regime migrations*

Description

This function calculates the generalized impulse response functions of an estimated MRVAR(n,p,S) for given a shock vector.

Usage

```
girf_MRGVAR_RM_CB(
  res,
  shock,
  R,
  nstep,
  Omega_hist = NA,
  resid_method = "parametric",
  conf,
  N
)
```

Arguments

res	: a MRGVAR object containing the components of the output of MRGVARData or MRGVARest.
shock	: an mn-vector containing the shocks as impulse.
R	: the number of runs to integrate out the random effects in order to obtain the means (see equation above).
nstep	: the length of the responses
Omega_hist	: the initial values from which the simulation runs of impulse and response functions start
resid_method	: resid_method = c("resid", "parametric"), It generate random residuals either from residuals bootstrap or parametric bootstrap.
conf	: a two component vector containing the tail probabilities of the bootstrap confidence interval.
N	: number of bootstrapping runs

Details

```
GIRF(shock=SHCK) = mean(Y(resid)) - mean(Y(SHCK))
```

It also generates the bootstrapped confidence intervals.

Value

an (n x n x nstep+1 x 3) array containing of impulse response functions with lower and upper confidence bonds. The rows represent response the columns represent impulses.

Examples

```
## case of n = 2, m = 2, S = 2     ## m: number of variables, n: number of countries
p = rep(1,12); dim(p) = c(2,3,2)
p[1,1,2] = 2; p[2,2,2]=2; p[,3,] = 0
TH = c(1:2)*0; dim(TH) = c(1,2)
res_d <- MRGVARData(m=2,n=2,p=p,TH=TH,T=100,S=2,SESVI=c(1,3),type="const")
max(res_d$Y)

### estimation of the MRGVAR model
colnames(res_d$Y) = c("P","Q","Pa","Qa")
res_e = MRGVARest(res=res_d)
res_e$Summary
```

```
Mod(STAT(res_e$Go[,,,1]))
Mod(STAT(res_e$Go[,,,2]))

GIRF    = girf_MRGVAR_RM(res=res_e,shock=c(1,1,1,1),R=100,nstep=10,Omega_hist=NA,
resid_method='parametric')
GIRF_CB = girf_MRGVAR_RM_CB(res=res_e,shock=c(1,1,1,1),R=100,nstep=10,Omega_hist=NA,
resid_method='parametric',conf=c(0.05,0.95),N=10)
GIRF_g  = IRF_graph(GIRF_CB,Names=c("P","Q","Pa","Qa"))
```

---

girf_MRVAR_RM_CB *Generalized impulse response functions of MRVAR(n,p,S) with regime migrations and confidence bands*

Description

This function calculates the generalized impulse response functions of an estimated MRVAR(n,p,S) for given a shock vector. It provides also a bootstrapped confidence interval.

Usage

```
girf_MRVAR_RM_CB(
  RES,
  shock,
  R,
  nstep,
  Omega_hist = NA,
  resid_method = "parametric",
  conf_level,
  N
)
```

Arguments

RES	: an estimated MRVAR object.
shock	: an n-vector containing the shocks as impulse.
R	: number of runs to integrate out the random effects in order to obtain the means (see equation above).
nstep	: length of the impuse response function
Omega_hist	: a (P x n) matrix of initial values, from which the impulse response functions start. Omega_hist determines from which regime the impulse response functions start.
resid_method	: resid_method = c("resid", "parametric"), It generate the random residuals from residuals bootstrap or parametric bootstrap.
conf_level	: a vector contain the level of confidences
N	: the number of bootstrap runs in containing the bootstrapped confidence intervals.

Details

GIRF(shock=SHCK) = mean(Y(resid)) - mean(Y(SHCK)): See H.H. Pesaran and Y. Shin (1998) Generalized impulse response analysis in linear multivariate models, Economics Letters, 58(1) p. 17-29. and G. Koop, M. H. Pesaran, and S. M. Potter (1996), Impulse response analysis in nonlinear multivariate models, Journal of Econometrics, 74 (1996) 119-74.

Value

an (n, n, nstep+1,3) array containing the impulse response functions with lower and upper confidence bonds. The rows represent response, and the columns represent impulses.

Examples

```
p = matrix(c(2,1,0,0),2,2)
res_d = MRVARData(n=2,p=p,T=300,S=2,SESVI=1,type="none")
max(res_d$Y)
colnames(res_d$Y) = c("R","P")
res_e = MRVARest(res=res_d)
RF3 = girf_MRVAR_RM(RES=res_e,shock=c(1,1),R = 200,nstep=20,Omega_hist=NA,resid_method="parametric")
RF4 = girf_MRVAR_RM_CB(RES=res_e, shock=c(1,1), R=200, nstep=20, Omega_hist=NA,
resid_method = "parametric", conf_level=c(0.05,0.95), N=100)
IRF_list <-IRF_graph(RF4)
```

GVARData — *Data generating process of GVAR(m,n,p)*

Description

This function generates data from a stationary global vector auto regressive process and return a GVAR(m,n,p) object containing data and parameters used in the GVAR(m,n,p) process.

Usage

```
GVARData(
  m,
  n,
  p,
  T,
  W = NA,
  r_npo = NA,
  Ao = NA,
  Bo = NA,
  Co = NA,
  Uo = NA,
  Sigmao = NA,
  type = NA,
  X = NA,
  mu = NA
)
```

Arguments

m	: number of variables
n	: number of countries/units
p	: an (n x 3) matrix in which each row contains the lag length of the domestic variables, the lag length of the foreign variables, and the number of exogenous variables.
T	: number of observations `(m,n,p,T) are parameters which must be provided.`
W	: an (n x n) weighting matrix. w_ij is the weight of foreign country j in the foreign variables of i-th country diag(W)=0
r_npo	: an (m, p, n) array collecting the roots of the characteristic polynomials in Lags for each of the m dimensional variable across n countries.
Ao	: an (m, m, p, n) array collecting the off-diagonal block of coefficients that are coefficients of the foreign variables.
Bo	: an (m, m, p, n) array collecting the coefficients of the domestic variables.
Co	: an (m, k+1, n) array collecting the coefficients of the deterministic components of the n countries.
Uo	: an (T x mn) matrix of the temporally independent innovations
Sigmao	: an (mn x mn) matrix of the covariance matrix of the GVAR(m,n,p) (W,r_npo,Ao,Bo,Uo,Sigmao) if not provided, they will be generated randomly.
type	: types of deterministic components: "const", "none", "exog0", and "exog1" are the options.
X	: (T x k x n) array of exogenous variables.
mu	: if type = "const" mu has the same dimension as Co. It contains the means of the time series in the system.

Value

a GVAR object that is a list containing the generated data, the used parameters, and the inputted of exogenous variables.

Examples

```
n = 5
p = (1:15)*0; dim(p) = c(5,3)
p[,1] = 2; p[,2]=1;
res_d = GVARData(m=2,n=5,p=p,T=100,type="const")
max(res_d$Y)
dim(res_d$Y)
res_e = GVARest(res = res_d)
res_e$Summary

X1 = matrix(1,200,1)
X2 = matrix(rnorm(200),200,1)
X3 = matrix(rnorm(200),200,1)
X4 = matrix(rnorm(200),200,1)
X  = cbind(X1,X2,X3,X4)
dim(X) = c(200,1,4)
n = 4
```

```
p = (1:12)*0; dim(p) = c(4,3);p[,1] = 2; p[,2]=1;   p[,3]=1; p[2,2]=2;
p    ## country-wise lag specification

res_d = GVARData(m=2,n=4,p=p,T=200,type="exog0",X=X)
res_e = GVARest(res = res_d)
res_e$Summary

IRF_CB = irf_GVAR_CB(res_e,nstep=10,comb=NA,irf="gen",runs=200,conf=c(0.05,0.95))
dim(IRF_CB)
IRF_g = IRF_graph(IRF_CB,Names=NA,response=c(1,4),impulse=c(1,2,3,4), ncol=4)
```

`GVARest` *Estimation of GVAR(m,n,p)*

Description

This function estimates the parameters of a specified GVAR(m,n,p) model based on provided data.

Usage

```
GVARest(res)
```

Arguments

`res`	: an GVAR object that is an output of GVARData including at least: m,n,p,type,Y and optionally X.

Value

res : an GVAR object with estimated parameter values, AIC, BIC, AIC_g, BIC_g and LH, where AIC and BIC are n-vectors of the country equations' AIC and BIC and AIG_g and BIC_g are the GVAR information criteria respectively.

Examples

```
n = 5
p = (1:15)*0; dim(p) = c(5,3)
p[,1] = 2; p[,2]=1;
res_d = GVARData(m=2,n=5,p=p,T=100,type="const")
max(res_d$Y)
dim(res_d$Y)
res_e = GVARest(res = res_d)
res_e$Summary

X1 = matrix(1,200,1)
X2 = matrix(rnorm(200),200,1)
X3 = matrix(rnorm(200),200,1)
X4 = matrix(rnorm(200),200,1)
X  = cbind(X1,X2,X3,X4)
dim(X) = c(200,1,4)
n = 4
p = (1:12)*0; dim(p) = c(4,3);p[,1] = 2; p[,2]=1;   p[,3]=1; p[2,2]=2;
p
```

```
res_d = GVARData(m=2,n=4,p=p,T=200,type="exog0",X=X)
res_e = GVARest(res = res_d)
res_e$Summary

IRF_CB = irf_GVAR_CB(res_e,nstep=10,comb=NA,irf="gen",runs=200,conf=c(0.05,0.95))
dim(IRF_CB)
IRF_g = IRF_graph(IRF_CB,Names=NA,response=c(1,4),impulse=c(1,2,3,4), ncol=4)
```

GVAR_Select *GVAR lag selection*

Description

Calculation of the information criteria of a GVAR country models for a given range of maximum lags.

Usage

```
GVAR_Select(res = res, L_V = L_V, I = I)
```

Arguments

res	: a GVAR object obtained from GVARData or GVARest.
L_V	: a two components vector containing the maxima of the domestic lag and the foreign lag, respectively.
I	: Index of the country under investigation.

Value

A matrix with different lag specifications and values of the model information criteria.

Examples

```
n = 4
p = (1:12)*0; dim(p) = c(4,3);p[,1] = 2; p[,2]=1; p[2:3,2] = 2
res_d = GVARData(m=2,n=4,p=p,T=4000,type="const")


I = 3
L_V = c(4,4)
res_d$p
GVARSelect = GVAR_Select(res=res_d,L_V=c(4,4),I=2)
GVARSelect[which.min(GVARSelect[,3]),]
GVARSelect[which.min(GVARSelect[,4]),]
```

irf_B_sigma *Impulse response function of a vector autoregressive model*

Description

This function generates impulse response functions for VAR,CIVAR,MRVAR MRCIVAR, also for GVAR, CIGVAR, MRGVAR and MRCIGVAR. For the later four classes of models it also provides the functionalities to calculate the global, regional and country-specific shocks. It also calculates global and regional responses and coordinated policy actions.

Usage

```
irf_B_sigma(
  B,
  sigma,
  nstep,
  comb,
  irf = c("gen", "chol", "chol1", "gen1", "genN1", "comb1", "smat", "concerts1",
    "PTdecomp", "ABSVAR", "irfX"),
  G = NA,
  A0 = NA,
  B0 = NA,
  smat = NA,
  Xshk = NA
)
```

Arguments

B	An (nxnxp) coefficients array of an n-dimensional VAR(p) model
sigma	The covariance matrix of the VAR(p) model
nstep	Number of steps of the impulse response function
comb	An n-vector of weights of the coordinated policy actions
irf	Type of impulse response function
G	The matrix used in the permanent and transitory decomposition
A0	The matrix for A/B identification in VAR
B0	The matrix for A/B identification in VAR
smat	An explicit decomposition matrix that defines a structural shock.
Xshk	The shock matrix of a one unit exogenous shock

irf_CIGVAR_CB *Impulse Response Functions of CIGVAR(m,n,p)*

Description

This function generates impulse response functions of an estimated CIGVAR(m,n,p) with confidence bands.

Usage

```
irf_CIGVAR_CB(
  res,
  nstep,
  comb,
  irf = c("gen", "chol", "chol1", "gen1", "comb1"),
  runs = 200,
  conf = c(0.05, 0.95)
)
```

Arguments

res	: a CIGVAR object of an output of CIGVARest.
nstep	: length of the impulse response functions
comb	: an mn-vector specifying combined impulse such as global shocks, regional shocks, or concerted actions.
irf	: types of the impulse response functions. irf=c("gen","chol","chol1","gen1","comb1"), gen for generalized IRF with one standard deviation shocks, gen1 for generalized IRF with one unit impulse, chol for IRF with Cholezky decomposition of the covariance matrix, chol1 for Cholezky decomposition with one unit impulse, comb1 for concerted action with one unit impulse.
runs	: number of bootstrap runs to generate the confidence bands
conf	: a two component vector containing the tail probabilities of the bootstrap confidence interval

Value

a matrix of (mn,mn,nstep,3) as the IRF columns representing the impulse rows the responses

Examples

```
n = 5
p = (1:15)*0; dim(p) = c(5,3)
p[,1] = 3; p[,2]=3;
p[2,1] = 2;  p[3,1] = 2
res_d = CIGVARData(m=2,n=5,p=p,T=500,type="const",DFYflag=0)
res_d$r_npo
res_d$check
res_e = CIGVARest(res_d)
res_e$Summary

### Impulse response function
```

```
IRF_CB = irf_CIGVAR_CB(res=res_e,nstep=20,comb=NA,irf="gen1",runs=200,conf=c(0.05,0.95))

dim(IRF_CB)
IRF_g = IRF_graph(IRF_CB,Names=NA,response=c(1,4),impulse=c(1,2,3,4), ncol=4)
```

---

irf_CIVAR_CB *Impulse response of an estimated CIVAR.*

Description

This function generates the impulse response functions of an estimated CIVAR model

Usage

```
irf_CIVAR_CB(
  res,
  nstep,
  comb,
  irf = c("gen", "chol", "chol1", "gen1", "comb1", "PTdecomp", "ABSVAR", "irfX"),
  G = NA,
  A0 = NA,
  B0 = NA,
  Xshks = NA,
  runs = 200,
  conf = c(0.05, 0.95)
)
```

Arguments

res	: a CIVAR object such as an output of CIVARest.
nstep	: the length of the impulse response functions.
comb	: a weighting matrix specifying the weights used in the impulse response functions of a global VAR. Its default value is NA for CIVAR(p).
irf	: types of the generated impulse response functions.
G	: the transformation matrix for PTdecomp
A0	: the transformation matrix for AB identification
B0	: the transformation matrix for AB identification
Xshks	: the number of selected exogenous shocks
runs	: number of runs used in the the calculation of the bootstrap confidence interval.
conf	: a two component vector containing the tail probabilities of the bootstrap confidence interval.

Value

an array of dimension (n, n, nstep, 3).

Examples

```
res_d = CIVARData(n=4,p=2,T=84,Co=matrix(c(1,1,1,1),4,1)*0,type="none",crk=1)
res_e = CIVARest(res=res_d)
res_e$Summary

IRF_CB = irf_CIVAR_CB(res=res_e, nstep=20, comb=NA, irf = "gen1", runs = 20, conf = c(0.05, 0.95))
IRF_g = IRF_graph(IRF_CB)

IRF_CB   = irf_CIVAR_CB(res=res_e, nstep=30, comb=NA, irf = "PTdecomp", G = NA, A0=NA,B0=NA,
 runs = 20, conf = c(0.05, 0.95))
IRF_g = IRF_graph(IRF_CB)
# The first three shocks have permanent effects,
# while the fourth shock does not have a permanent effect.



T=100
X=matrix(rnorm(2*T),T,2)
res_d = VARData(n=3,p=2,T=T,Co=matrix(c(0,0,0,1,2,3,3,2,1),3,3), type="exog0",X=X) ;
res_e = VARest(res=res_d);
res_d$Co
res_e$Summary
IRF_CB = irf_VAR_CB(res=res_e,nstep=20, comb=NA, irf = "irfX", Xshks=c(1:2),
runs = 100, conf = c(0.05, 0.95))
IRF_list <-IRF_graph(IRF_CB,Names =c("Y1","Y2","Y3"),INames=c("X1","X2"),
response = c(1:3), impulse = c(1:3), n = 3)
```

`irf_GloabalResponse_CB` *Global Responses*

Description

This function calculates global or regional responses from a set of bootstrapped impulse response functions and a weighting matrix for the aggregation of the global responses.

Usage

```
irf_GloabalResponse_CB(IRF_CB, comb_all)
```

Arguments

`IRF_CB`	an output of irf_MRGVAR_CB
`comb_all`	a weighting matrix for the aggregation of the global responses

Value

a list containing the global impulse response functions and the accumulated global impulse response functions.

irf_GVAR_CB *Impulse Response Functions of GVAR*

Description

This function generates impulse response functions of an estimated GVAR with confidence bands

Usage

```
irf_GVAR_CB(
  res,
  nstep,
  comb,
  irf = c("gen", "chol", "chol1", "gen1", "comb1"),
  runs = 200,
  conf = c(0.05, 0.95)
)
```

Arguments

res	: an object of GVAR that is a list of the output of GVARest
nstep	: length of the impulse response functions
comb	: an mn vector specifying combined impulse such as global shocks, regional shocks, or concerted actions.
irf	: types of the impulse response functions. irf=c("gen","chol","chol1","gen1","comb1"), gen for generalized IRF with one standard deviation shocks, gen1 for generalized IRF with one unit impulse, chol for IRF with Cholezky decomposition of the covariance matrix, chol1 for Cholezky decomposition with one unit impulse, comb1 for concerted action with one unit impulse.
runs	: number of bootstrap runs to generate the confidence bands
conf	: a two components vector of the tail probabilities of the confidence interval.

Value

An (mn,mn,nstep,3) array of IRF with columns representing the impulse rows the responses.

Examples

```
n = 5
p = (1:15)*0; dim(p) = c(5,3)
p[,1] = 2; p[,2]=1;
res_d = GVARData(m=2,n=5,p=p,T=100,type="const")
max(res_d$Y)
dim(res_d$Y)
res_e = GVARest(res = res_d)
res_e$Summary

IRF_CB = irf_GVAR_CB(res_e,nstep=10,comb=NA,irf="gen",runs=200,conf=c(0.05,0.95))
dim(IRF_CB)
IRF_g = IRF_graph(IRF_CB,Names=NA,response=c(1,4),impulse=c(1,2,3,4), ncol=4)
```

irf_MRCIGVAR_CB *Regime specific impulse response functions of an MRCIG-VAR(m,n,p,S) model*

Description

This function calculates the regime specific impulse response functions of an estimated MRCIG-VAR(n,p,S,r). Using G[,,,s] and Sigma[,,s] matrices of the estimated MRCIGVAR, this function can produce impulse response functions for any possible combinations of states.

Usage

```
irf_MRCIGVAR_CB(
  res,
  nstep,
  comb,
  state = c(2, 1),
  irf = c("gen", "chol", "chol1", "gen1", "comb1"),
  runs = 200,
  conf = c(0.05, 0.95),
  NT = 1
)
```

Arguments

res	a MRCIGVAR object that can be an output of MRCIGVARData, MRCIGVARest, or MRCIGVARest.
nstep	the length of impulse response function
comb	a vector specify the concerted action in policy-simulation impulse response function
state	an n-vector specifying the specific state for each country.
irf	types of the impulse response function c("gen","chol","chol1","gen1","comb1")
runs	the number of bootstrapping runs
conf	A vector specifying the confidence intervals
NT	The number of impulse response scenarios

Value

an list of an (mn,mn,nstep,3,S) array of the impulse response functions and test statistics. In the impulse response array, columns representing the impulse and rows the responses.

Examples

```
m = 2
n = 3
p = c(2,2,2,2,2,2,2,2,2,2,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,0,0,0,0,0); dim(p) = c(5,3,2)
p = p[1:n,,]; p[,1,] = 3; p[,2,] = 2

TH = c(1:n)*0; dim(TH) = c(1,n)
SESVI=rep(1,3,5)
```

```
r  = rep(1,n)

## case of n = 3, m = 2, S = 2    m: number of variables, n: number of countries


res_d <- MRCIGVARData(m=2,n=3,p=p,TH=TH,T=100,S=2, SESVI=c(1,3,5),r=rep(1,3),Ncommtrend=1)
max(abs(res_d$Y))
STAT(res_d$Go[,,,2])
STAT(res_d$Go[,,,1])
res_e  = MRCIGVARest(res=res_d)

res_e$Summary

IRF_CB = irf_MRCIGVAR_CB(res=res_e, nstep=10, comb=NA, state = c(2,1,1), irf = "gen1",runs=20,
 conf = c(0.05, 0.95), NT = 1)

str(IRF_CB)

IRF_g = IRF_graph(IRF_CB[[1]])
```

`irf_MRCIVAR_CB` *Regime specific impulse response functions of MRCIVAR(n,p,S)*

Description

This function calculates the regime specific impulse response functions with confidence bands, using Bo[,,,s] and Sigma[,,s] matrices of the estimated MRCIVAR.

Usage

```
irf_MRCIVAR_CB(
  res_e,
  nstep = 20,
  irf = c("gen", "gen1"),
  runs = 100,
  comb = NA,
  G = NA,
  smat = NA,
  conf = c(0.05, 0.95)
)
```

Arguments

`res_e`	an object of MRCIVAR as output of MRVARestm
`nstep`	the length of impulse response function
`irf`	types of the impulse response function c("gen","chol","chol1","gen1","comb1"), gen for GIRF, gen1 for GIRF with unit impulse, chol Cholezky decomposition, chol1 Cholezky decomposition with unit impulse, comb1 concerted action with a one unit impulse.
`runs`	Number of simulation runs

comb	a vector specify the concerted action in policy-simulation impulse response function
G	The matrix used in the permanent and transitory decomposition
smat	An explicit decomposition matrix that defines a structural shock.
conf	a vector of the tail probabilities of the confidence interval.

Value

A list of impulse response function in two regimes and the bootstrap parameters.

Examples

```
n =10

Sigma = 1:(n*n*2)
dim(Sigma) = c(n,n,2)
Sigma[,,1] = diag(n)
Sigma[,,2] = diag(n)
p=matrix(0,2,2)
p[,1] = c(3,3)
res_d = MRCIVARDatam(n=n,p=p,T=250,S=2,SESVI=1,TH=0,Sigmao=Sigma,type="const",r=2)
#colnames(res_d$Y) = c("w","p","I","Q")
max(abs(res_d$Y))
res_e = MRCIVARestm1(res=res_d)
#res_e$Summary
if (! max(Mod(STAT(res_e$Bo[,,,1])),Mod(STAT(res_e$Bo[,,,2])) ) > 1.0001 ) {

 IRF  = irf_MRCIVAR_CB(res_e,nstep=20,irf="gen1",runs=100,comb=NA,G=NA,conf=c(0.05,0.95))
  IRF_g1 <- IRF_graph(IRF[[1]])        # irf of regime 1
  IRF_g2<- IRF_graph(IRF[[2]])         # irf of regime 2

}
```

---

irf_MRGVAR_CB   *Regime specific impulse response functions of MRGVAR(n,p,S)*

Description

This function calculates the regime specific impulse response functions of an estimated MRGVAR(n,p,S). Using the estimated G[,,,s] and Sigma[,,s] matrices of the MRGVAR, this function calculated the regime specific impulse response functions.

Usage

```
irf_MRGVAR_CB(
  res,
  state = c(2, 1),
  nstep,
  comb,
  irf = c("gen", "chol", "chol1", "gen1", "comb1"),
  G = NA,
```

```
    smat = NA,
    sigmaNPDS = NA,
    runs = 200,
    conf = c(0.05, 0.95),
    NT = 1
)
```

Arguments

`res`	a list of estimated MRGVAR as output of MRGVARest
`state`	an n vector specifying the specific state for each country.
`nstep`	the length of impulse response function
`comb`	a vector specify the concerted action in policy-simulation impulse response function
`irf`	: types of the impulse response irf=c("gen","chol","chol1","gen1","comb1") "gen" for generalized impulse response with one standard deviation impulses, "gen1" for GIRF with one unit impulses, "chol" Cholezky decomposition, "chol1" Cholezky decomposition with unit impulses, "comb1" concerted action with unit impulse.
`G`	For permanent and transitory decomposition
`smat`	For an explicit structural decomposition of the correlated shocks
`sigmaNPDS`	the state-dependent covariance matrix
`runs`	number of bootstrapping runs
`conf`	A vector containing confidence levels
`NT`	number of impulse response scenarios in a simulation run

Value

a list of bootstrap result. The first component contains the impulse response functions with confidence bands. It is an (mn,mn,nstep,3) array where the IRF columns represent the impulse and rows represent the responses.

Examples

```
## case of n = 2, m = 2, S = 2     ## m: number of variables, n: number of countries
p = rep(1,12); dim(p) = c(2,3,2)
p[1,1,2] = 2; p[2,2,2]=2; p[,3,] = 0
TH = c(1:2)*0; dim(TH) = c(1,2)
res_d <- MRGVARData(m=2,n=2,p=p,TH=TH,T=100,S=2,SESVI=c(1,3),type="const")
max(res_d$Y)

### estimation of the MRGVAR model
colnames(res_d$Y) = c("P","Q","Pa","Qa")
res_e = MRGVARest(res=res_d)
res_e$Summary

IRF_CB  = irf_MRGVAR_CB(res=res_e,nstep=10,comb=NA,state=c(1,1),irf="gen1",runs=20,
conf=c(0.05,0.95))
IRF_g = IRF_graph(IRF_CB[[1]],Names=c("P","Q","Pa","Qa"))    #IRF
#IRF_g = IRF_graph(IRF_CB[[2]])   # accumulated IRF
```

irf_MRVAR_CB *Regime specific impulse response functions of MRVAR(n,p,S)*

Description

This function calculates the regime specific impulse response functions of an estimated MRVAR(n,p,S).

Usage

```
irf_MRVAR_CB(
  res_e,
  nstep,
  comb,
  irf = c("gen", "chol", "chol1", "gen1", "comb1", "irfX"),
  Xshks = NA,
  runs = 200,
  conf = c(0.05, 0.95)
)
```

Arguments

res_e	an output of MRVARest.
nstep	the length of impulse response function
comb	a vector specify the concerted action in policy-simulation impulse response function
irf	types of the impulse response functions. irf=c("gen","chol","chol1","gen1","comb1"), gen for generalized IRF with one standard deviation shocks, gen1 for generalized IRF with one unit impulse, chol for IRF with Cholezky decomposition of the covariance matrix, chol1 for Cholezky decomposition with one unit impulse, comb1 for concerted action with one unit impulse in GVAR.
Xshks	number of selected exogenous shocks
runs	number of simulation runs
conf	a two components vector containing the tail probabilities of the bootstrap confidence interval.

Value

an (n,n,nstep,3,2) array containing the IRF of the two regimes. The IRF columns represent the impulse and the rows the responses.

Examples

```
p = matrix(c(2,1,0,0),2,2)
res_d = MRVARData(n=2,p=p,T=300,S=2,SESVI=1)
max(abs(res_d$Y))
res_e  = MRVARest(res_d)
res_e$Summary


IRF_CB = irf_MRVAR_CB(res_e,nstep=20,comb=NA,irf="gen1",runs=100,conf=c(0.05,0.90))
```

```
IRF_list1 <-IRF_graph(IRF_CB[[1]])
IRF_list2 <-IRF_graph(IRF_CB[[2]])
```

`irf_MRVAR_NM_CB` *Regime specific impulse response functions of MRVAR(n,p,S)*

Description

This function calculates the regime specific impulse response functions of an estimated MRVAR(n,p,S), using Bo[,,,s] and Sigma[,,s] of the estimated MRVAR.

Usage

```
irf_MRVAR_NM_CB(
  RESS,
  nstep,
  comb,
  irf = c("gen", "chol", "chol1", "gen1", "comb1"),
  runs = 200,
  conf = c(0.05, 0.95)
)
```

Arguments

RESS	an output of MRVARest.
nstep	the length of impulse response function
comb	a vector specify the concerted action in policy-simulation impulse response function
irf	types of the impulse response functions. irf=c("gen","chol","chol1","gen1","comb1"), gen for generalized IRF with one standard deviation shocks, gen1 for generalized IRF with one unit impulse, chol for IRF with Cholezky decomposition of the covariance matrix, chol1 for Cholezky decomposition with one unit impulse, comb1 for concerted action with one unit impulse in GVAR.
runs	number of simulation runs
conf	a two components vector containing the tail probabilities of the bootstrap confidence interval.

Value

an (n,n,nstep,3,2) array containing the IRF of the two regimes. The IRF columns represent the impulse and the rows the responses.

Examples

```
p = matrix(c(2,1,0,0),2,2)
res_d = MRVARData(n=2,p=p,T=300,S=2,SESVI=1)
max(abs(res_d$Y))
res_e  = MRVARest(res_d)
res_e$Summary
```

```
IRF    = irf_MRVAR_NM(res_e,nstep=10,comb=NA,irf="gen")
IRF_CB = irf_MRVAR_NM_CB(res_e,nstep=10,comb=NA,irf="gen",runs=200,conf=c(0.05,0.90))
IRF_list1 <-IRF_graph(IRF_CB[,,,,1])
IRF_list2 <-IRF_graph(IRF_CB[,,,,2])
```

---

irf_VAR_CB *Impulse response function of VAR(p)*

Description

This function generates impulse response functions of an estimated VAR(p)

Usage

```
irf_VAR_CB(
  res,
  nstep,
  comb,
  irf = c("gen", "chol", "chol1", "gen1", "comb1", "irfX"),
  runs = 200,
  conf = c(0.05, 0.95),
  smat = NA,
  Xshks = NA
)
```

Arguments

res	an object of the output of VARest
nstep	length of the impulse response functions
comb	a vector containing the weights of a combined policy actions
irf	types of impulse response functions
runs	number of simulation runs in generating the confidence interval of the impulse response functions
conf	confidence levels
smat	chock to response transformation matrix for the option irf="smat"
Xshks	the number of exogenous variables

Value

an (n, n, nstep, 3) array containing the impulse response functions

Examples

```
res_d = VARData(n=2,p=2,T=100,type="const")
res_e = VARest(res=res_d);
IRF_CB = irf_VAR_CB(res=res_e,nstep=20, comb=NA, irf = "gen1", runs = 100, conf = c(0.05, 0.95))
IRF_list <-IRF_graph(IRF_CB)
res_e$Summary
```

MIxCIVARData *Mixed VECM with I(0) and I(1) variables*

Description

This function generates and estimates a mixed VECM with I(0) and I(1) variables

Usage

```
MIxCIVARData(
  n,
  p,
  T,
  r,
  k,
  type,
  Bo = NA,
  Y = NA,
  X = NA,
  D = NA,
  Go = NA,
  B = NA,
  Sigma = NA
)
```

Arguments

n	: dimension of the mixed CIVAR
p	: lag of the mixed CIVAR
T	: number of observations or length of the generated data
r	: number of unit roots in the joint CIVAR
k	: number of I(0) variables
type	: types of the deterministic components in the conditional CIVAR
Bo	: coefficient matrix of the mixed CIVAR
Y	: data of the mixed CIVAR
X	: data of the exogenous variables
D	: transformation matrix to mix the I(0) and I(1) components
Go	: pre-loaded selection matrix
B	: coefficient matrix of the mixed CIVAR
Sigma	: covariance matrix of the residuals

Value

a list contains estimation and test results of the mixed VECM

Examples

```
#RR <- MIxCIVARData(n=9,p=2,T=209,r=5,k=2,type="const",Bo=NA,Y=NA,D=NA)
## DGP of I(1) and I(0) mixed VECM
#plot(ts(RR$Y))
#
#### testing the mixed VECM via testing the restrictions on beta
#res_d <- CIVARData(n=9,p=2,T=209,type="const",crk=4)
#res_e = CIVARest(res=res_d)
#res_e$Summary
#n = 9; crk = 4; k = 2; r = 5
#CC  <- c(8,9,17,18)
#GG  <- c(19:25,28:34)
#G = diag(n*crk); psi=matrix(1,n*crk,1)
#### this implies there is no restrictions on the adjustment coefficients alpha
#H = diag(n*crk);          H2 = H[,-c(seq(1,(n-r-k)*n,n),seq((n-r-k+1)*n,(n-r)*n,n),CC,GG)]
#### only normalization
#h = matrix(0,n*crk,1);        h[c(seq(1,(n-r-k)*n,n),seq((n-r-k+1)*n,(n-r)*n,n)),1] <- 1
#phi = matrix(1,ncol(H2),1)
### check consistency of the restrictions    G%*%psi; H2%*%phi + h
#
#G%*%psi; H2%*%phi + h
#
#ABtest = CIVARTest(res=res_d,H=H2,h=h,phi=phi,G=G,Dxflag=0)
#ABtest$betar
#ABtest$alphar
#ABtest$VECMR$coefficients
#ABtest$LR
#1-pchisq(ABtest$LR,14)   ### The Ho of last two are I(0) is rejected
#
#res_d$Y = RR$Y           ### replacing the data of two I(0) variables
#ABtest = CIVARTest(res=res_d,H=H2,h=h,phi=phi,G=G,Dxflag=0)
#ABtest$betar
#ABtest$alphar
#ABtest$VECMR$coefficients
#ABtest$LR
#1-pchisq(ABtest$LR,14)  ### The Ho of last two are I(0) cannot be rejected
#
#RR$GABtest$LR           ## RR contains the same LR
#
#
```

`MRCIGVARData` *Data generating process of MRCIGVAR(m,n,p,S,r)*

Description

This function generates data from an multi-regime cointegrated global VAR process and returns an MRCIGVAR(m,n,p,S) object that is a list containing the data and the parameters used in the MRCIGVAR(m,n,p,S) process.

Usage

```
MRCIGVARData(
  m,
```

```
  n,
  p,
  T,
  S,
  W = NA,
  SESVI = NA,
  TH = NA,
  Go = NA,
  Ao = NA,
  Bo = NA,
  Sigmao = NA,
  Uo = NA,
  SV = NA,
  type = NA,
  Co = NA,
  X = NA,
  Yo = NA,
  d = NA,
  r = NA,
  r_np = NA,
  Ncommtrend = NA,
  DFYflag = 0,
  A = NA,
  Ncommfakt = 1,
  uz = NA
)
```

Arguments

m	: number of variables in each country/unit
n	: number of countries/units
p	: an (n, 3, S) array, each raw contains the lag length of the domestic variables, the lag length of the foreign variables, and the number of the exogenous variables of the corresponding country in the respective regimes.
T	: number of observations.
S	: number of regimes
W	: an (n x n) weighting matrix. w_ij is the weight of country j in the foreign variables of i-th country diag(W)=0
SESVI	: an n-vector of indices of the switching variables across n countries. E.g. SESVI = seq(1,m*n,m).
TH	: an (n x S-1) matrix of threshold values
Go	: an (mn,mn,p,S) array containing coefficients of MRCIGVAR. Go is constructed from Bo, Ao and W.
Ao	: an (m, m, p, n, S) array containing the coefficients of foreign variables
Bo	: an (m, m, p, n, S) array containing the coefficients of domestic variables.
Sigmao	: an (mn, mn, S) array of the covariance matrix of MRCIGVAR(m,n,p,S)
Uo	: a (T, mn, S) array of the temporally independent innovations
SV	: exogenous switching variables

type	: types of the deterministic component. "const", "none", "exog0", and "exog1" are 4 options
Co	: an (m , k+1, n, S) array containing the coefficients of the deterministic components of the n countries.
X	: a (T x k x n x S) array of exogenous stationary variables.
Yo	: initial values
d	: the time lag between signal and regime-switching
r	: an n-vector containing the number of unit root process in each country
r_np	: an (m, Pmax, n, S) array containing the roots of the characteristic polynomials of each country.
Ncommtrend	: number of common stochastic trends in the MRCIGVAR.
DFYflag	: indicator whether the foreign variables enter the cointegration space.
A	: transformation matrix for common exogenous non-stationary stochastic factors.
Ncommfakt	: number of the common exogenous non-stationary stochastic factors.
uz	: innovations of the common exogenous non-stationary stochastic factors.

Value

an MRGVAR object containing the generated data, the used parameters and the exogenous variables.

- Y : a (T x mn) matrix of simulated data
- X : a (T x k x n x S) array of exogenous variables
- Uo : a (T, mn,S) array of the simulated innovations of the MRGVAR(m,n,p,S)
- C : an (nm, (k+1),S) array containing the coefficients of the deterministic components.
- St : a (T x n) matrix of the simulated time path of states/regimes
- check : maximum of the data for checking the stationarity

Examples

```
m = 3
n = 5
 p = c(2,2,2,2,2,2,2,2,2,2,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,0,0,0,0,0); dim(p) = c(5,3,2)
p = p[1:n,,]; p[,1,] = 3; p[,2,] = 2

p[2,2,] = 3
TH = c(1:n)*0; dim(TH) = c(1,n)
SESVI=rep(1,4,7,10,13)
r  = rep(1,n)
i = 1
S = 2

T= 200
XX = matrix(rnorm(T*10*2),T*10,2)
dim(XX) = c(T,2,5,2)
p[,3,]=2

res_d <- MRCIGVARData(m=3,n=5,p=p,TH=TH,T=T,S=2, SESVI=c(1,4,7,10,13),type="exog0",r=rep(1,5),
DFYflag=0,Ncommfakt=1,X=XX)    ## m: number of variables, n: number of countries
```

```
max(abs(res_d$Y))
plot(ts(res_d$Y[,1:10]))
STAT(res_d$Go[,,,2])
STAT(res_d$GC[,,,1]);STAT(res_d$GC[,,,2]);
STAT(res_d$Go[,,,1])
max(abs(res_d$Y))

Ao=res_d$Ao;Bo=res_d$Bo;Co=res_d$Co;

res_d <- MRCIGVARData(m=3,n=5,p=p,TH=TH,T=T,S=2, SESVI=c(1,4,7,10,13),type="exog0",Ao=Ao,Bo=Bo,
Co=Co,r = rep(1,5),DFYflag = 0,Ncommfakt=1,X=XX) ##m:number of variables, n:number of countries
res_e  = MRCIGVARest(res=res_d)
```

MRCIGVARest *Estimation of MRCIGVAR(res)*

Description

This function estimates parameters of a multi regime cointegrated global VAR(p) model. The adjustment speeds to the cointegration relations are assumed to be different in different regimes.

Usage

```
MRCIGVARest(res)
```

Arguments

res	: an MRCIVAR object of the output of MRCIGVARData

Value

: an MRCIVAR object with estimated parameters and test statistics.

Examples

```
m = 2
n = 3
p = c(2,2,2,2,2,2,2,2,2,2,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,0,0,0,0,0); dim(p) = c(5,3,2)
p = p[1:n,,]; p[,1,] = 3; p[,2,] = 2

TH = c(1:n)*0; dim(TH) = c(1,n)
SESVI=rep(1,3,5)
r  = rep(1,n)

## case of n = 3, m = 2, S = 2     ## m: number of variables, n: number of countries

res_d <- MRCIGVARData(m=2,n=3,p=p,TH=TH,T=300,S=2, SESVI=c(1,3,5),r=rep(1,3),Ncommtrend=1)
# Ao
max(abs(res_d$Y))
plot(ts(res_d$Y))
STAT(res_d$Go[,,,2])
STAT(res_d$Go[,,,1])
```

```
max(abs(res_d$Y))
res_d$type
res_d$W
#   Ao =  res_d$Ao;  Bo =  res_d$Bo;  Co =  res_d$Co;
#   res_d <- MRCIGVARData(m=2,n=3,p=p,TH=TH,T=300,S=2, SESVI=c(1,3,5),Ao=Ao,Bo=Bo,Co=Co)

max(res_d$Y)
res_e  = MRCIGVARest(res=res_d)
res_e$Summary
```

`MRCIGVAR_Select` *Calculation of information criteria AIC and BIC for an estimated MRCIGVAR model*

Description

Calculation of information criteria AIC and BIC for an estimated MRCIGVAR model

Usage

```
MRCIGVAR_Select(res, L_V = L_V, TH_V = TH_V)
```

Arguments

`res`	: an MRCIGVAR object generated from MRCIGVARData or estimated from MRCIGVARest.
`L_V`	: a 2 components vector containing the maxima of the domestic lag length and the foreign lag length respectively.
`TH_V`	: a vector containing possible threshold values .

Value

a matrix with different lag specifications, threshold values and the corresponding information criteria.

Examples

```
m = 2                  ## m: number of variables, n: number of countries
n = 4
p = c(2,2,2,2,2,2,2,2,2,2,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,0,0,0,0,0); dim(p) = c(5,3,2)
p = p[1:4,,]; p[,1:2,2] = 3

TH = c(1:4)*0; dim(TH) = c(1,4)

res_d <- MRCIGVARData(m=2,n=4,p=p,TH=TH,T=400,S=2, SESVI=c(1,3,5,7),r=rep(1,4))

res_em = MRCIGVARest(res=res_d)

L_v = c(3,3)
TH_v = c(0,0.1)
res_d$p
```

```
MRCIGVARSelect = MRCIGVAR_Select(res=res_d,L_V=L_v,TH_V=TH_v)
dim(MRCIGVARSelect)
MRCIGVARSelect[which.min(MRCIGVARSelect[,28]),]
```

MRCIVARDatam *Data generating process of MRCIVAR(n,p,S)*

Description

This function generates data from an multi-regime cointegrated VAR(p) process with identical cointegration relations and different adjustment speeds and returns a MRCIVAR which is a list containing data and parameters used in the multi regime cointegrated VAR(p) process.

Usage

```
MRCIVARDatam(
  n = 2,
  p = matrix(2, 2, 2),
  T = 100,
  S = 2,
  SESVI,
  TH,
  Bo,
  Co,
  Sigmao,
  Uo,
  SV,
  type,
  X,
  mu,
  Yo,
  Do,
  d = 1,
  r = 1
)
```

Arguments

n	: number of variables
p	: an (S x 2) matrix. Each row of p contains the lag length for the corresponding regime and the number of exogenous variables for the regime.
T	: number of observations
S	: number of regimes
SESVI	: index of the switching variable, switching sv_t = Y[t-1,SESVI] > Y[t-2,SESVI] `(n,S,p,T,SESVI) are parameters that have to be provided.`
TH	: an (S-1) vector of threshold values
Bo	: an (n,n,p,S) array of the coefficients of MRVAR(n,p,S). If not given it will be generated.

Co	: an (n,k+1,S) array of the coefficients of the deterministic components. For type="none" Co = O*(1:n,1:S), for "const" Co is an n-vector for each regime, "exog0" Co is a (n,k+1,S) array with first column of zeros for each regime respectively,for "exog1" Co is an (n,1+k, S) array without zero restrictions.
Sigmao	: an (n,n,S) array containing S covariance matrices of the residuals
Uo	: residuals, if it is not NA it will be used as input to generate the MRVAR(n,p,S) for each regime.
SV	: exogenous switching variable
type	: type of the deterministic components type = ("none","const","exog0","exog1")
X	: a (T x k x S) matrix of exogenous variables for each state. The second dimension can be filled with zeros to take into account that the the exogenous variables are not identical in each state.
mu	: an (n x S) matrix of the regime specific mean of the variables
Yo	: a (p, n, S) array of initial values of the process
Do	: a (T, n, S) array of extra exogenous components (not used with value zero)
d	: lag delay of the self-exiting switching
r	: number of unit roots in the system. (n - r) is then the cointegration rank.

`(TH,Bo,Sigmao,Uo,SV) if not provided, they will be generated randomly.`

Value

An MRCIVAR object containing the generated data, the used parameters, and the exogenous variables.

Examples

```
Sigma = 1:(4*4*2)
dim(Sigma) = c(4,4,2)
Sigma[,,1] = diag(4)
Sigma[,,2] = diag(4)
p=matrix(0,2,2)
p[,1] = c(3,3)
res_d = MRCIVARDatam(n=4,p=p,T=250,S=2,SESVI=1,TH=0,Sigmao=Sigma,type="const",r=2)
colnames(res_d$Y) = c("w","p","I","Q")
max(abs(res_d$Y))
res_e = MRCIVARestm1(res=res_d)
res_e$Summary

p=matrix(0,2,2)
p[,1] = c(3,3)
p[,2] = 1
T = 200
XX = matrix(rnorm(T),T,1)
XX = cbind(XX,XX); dim(XX) = c(T,1,2)
dim(XX)

res_d = MRCIVARDatam(n=4,p=p,T=T,S=2,SESVI=1,TH=0,Sigmao=Sigma,type="exog0",r=1,X=XX)
res_e = MRCIVARestm1(res=res_d)
res_e$Summary

res_d = MRCIVARDatam(n=4,p=p,T=T,S=2,SESVI=1,TH=0,Sigmao=Sigma,type="exog1",r=1,X=XX)
```

```
res_e = MRCIVARestm1(res=res_d)
res_e$Summary
```

MRCIVARestm1 *Estimation of MRCIVAR models*

Description

This function executes a Johansen test of cointegration ranks and estimates the parameters of a MRCIVAR(n,p,S) model.

Usage

```
MRCIVARestm1(res)
```

Arguments

res	an MRCIVAR object of the output of MRCIVARDatam

Value

an estimated MRCIVAR object containing estimated parameters and test results

Examples

```
Sigma = 1:(4*4*2)
dim(Sigma) = c(4,4,2)
Sigma[,,1] = diag(4)
Sigma[,,2] = diag(4)
p=matrix(0,2,2)
p[,1] = c(3,3)

res_d = MRCIVARDatam(n=4,p=p,T=250,S=2,SESVI=1,TH=0,Sigmao=Sigma,type="const",r=2)
colnames(res_d$Y) = c("w","p","I","Q")
max(abs(res_d$Y))
res_e = MRCIVARestm1(res=res_d)
res_e$Summary
```

MRCIVAR_Selectm *Calculation of information criteria AIC and BIC for MRCIVAR models*

Description

Calculation of information criteria AIC and BIC for MRCIVAR models

Usage

```
MRCIVAR_Selectm(res = res, L_V = L_V, TH_V = TH_V)
```

Arguments

`res` : an MRCIVAR object generated from MRCIVARDatam or estimated from MRCIVARestm1.

`L_V` : a 2 components vector containing the maxima of the lags length of the two regimes, respectively.

`TH_V` : a vector containing possible threshold values for the selection.

Value

A matrix with different lag specifications, threshold values, and the corresponding model selection criteria.

Examples

```
Sigma = 1:(4*4*2)
dim(Sigma) = c(4,4,2)
Sigma[,,1] = diag(4)
Sigma[,,2] = diag(4)
p=matrix(0,2,2)
p[,1] = c(3,2)

res_d = MRCIVARDatam(n=4,p=p,T=2610,S=2,SESVI=1,TH=0,Sigmao=Sigma,type="const",r=1)
res_e = MRCIVARestm1(res=res_d)

TH_v = c(0,0.1)
L_v = c(6,6)

Selm = MRCIVAR_Selectm(res=res_e,L_V=L_v,TH_V=TH_v)
MRVAR_Select_Summary(Selm)
```

MRGVARData *Data generating process of MRGVAR(m,n,p,S)*

Description

This function generates data from an multi-regime stationary global VAR(p) process and returns an MRGVAR object that is a list containing the generated data and the used parameters.

Usage

```
MRGVARData(
  m,
  n,
  p,
  T,
  S,
  W = NA,
  SESVI = NA,
  TH = NA,
  Go = NA,
```

```
  Ao = NA,
  Bo = NA,
  Sigmao = NA,
  Uo = NA,
  SV = NA,
  type = NA,
  Co = NA,
  X = NA,
  Yo = NA,
  d = NA
)
```

Arguments

`m`	: number of variables in a country/unit
`n`	: number of countries/units
`p`	: an (n, 3, S) array, each raw specifies the lag length of the domestic variables, the lag length of the foreign variables and the number of exogenous variables for the respective regime.
`T`	: number of observations.
`S`	: number of regimes.
`W`	: an (n x n) weighting matrix. w_ij is the weight of country j in the foreign variables of i-th country diag(W)=0
`SESVI`	: an n-vector of indices of the switching variables across n countries. Eg. SESVI = seq(1,m*n,m).
`TH`	: an (n, S-1) matrix of threshold values
`Go`	: an (mn,mn,p,S) array of the MRGVAR(m,n,p,S) coefficients. G is constructed from Bo, Ao and W.
`Ao`	: an (m, m, p, n, S) array collecting coefficients of foreign variables
`Bo`	: an (m, m, p, n, S) array collecting coefficients of domestic variables
`Sigmao`	: an (mn, mn, S) array of the covariance matrix of MRGVAR(m,n,p,S)
`Uo`	: a (T, mn, S) array of the temporally independent innovation processes
`SV`	: exogenous switching variables
`type`	: types of deterministic component "const", "none", "exog0", and "exog1" are 4 options
`Co`	: an (m , k+1, n, S) array collecting the coefficients of the deterministic components of the n countries for every regime.
`X`	: a (T x k x n x S) matrix of exogenous variables.
`Yo`	: Initial values
`d`	: the time lag between signal and switching

Value

an MRGVAR object containing the generated data, the used parameters and the exogenous variables. res_d = list(Y,X,Uo,resid,Go,GDC,Co,Sigmao,TH,St,SV,SESVI,Ao,Bo,check,type,m,n,p,S,W,SigmaS,Yo,d)

- Y : a (T x nm) matrix of simulated data via of the MRGVAR(m,n,p,S)

- X : a (T x k x n x S) matrix of exogenous variables.
- Uo : a (T, mn,S) array of the simulated innovations of the MRGVAR(m,n,p,S)
- C : an (nm, (k+1),S) array containing the coefficients of the deterministic components.
- St : a (T x n) matrix of the simulated time path of states/regimes
- check : maximum of the data for checking the stationarity

Examples

```
## case of n = 2, m = 2, S = 2     ## m: number of variables, n: number of countries
p = rep(1,12); dim(p) = c(2,3,2)
p[1,1,2] = 2; p[2,2,2]=2; p[,3,] = 0
TH = c(1:2)*0; dim(TH) = c(1,2)
res_d <- MRGVARData(m=2,n=2,p=p,TH=TH,T=100,S=2,SESVI=c(1,3),type="const")
max(res_d$Y)

### estimation of the MRGVAR model
colnames(res_d$Y) = c("P","Q","Pa","Qa")
res_e = MRGVARest(res=res_d)
res_e$Summary

IRF_CB  = irf_MRGVAR_CB(res=res_e,nstep=10,comb=NA,state=c(1,1),irf="gen1",runs=20,
conf=c(0.05,0.95))
IRF_g = IRF_graph(IRF_CB[[1]],Names=c("P","Q","Pa","Qa"))    #IRF
#IRF_g = IRF_graph(IRF_CB[[2]])   # accumulated IRF
```

`MRGVARest` *Estimation of MRGVAR(m,n,p,S) models*

Description

This function estimates the parameters of a specified MRGVAR(m,n,p,S) model based on provided data.

Usage

```
MRGVARest(res)
```

Arguments

`res`	: an MRGVAR object which is an output of MRGVARData.

Value

res : an MRGVAR object with estimated parameters and test statistics.

Examples

```
## case of n = 2, m = 2, S = 2     ## m: number of variables, n: number of countries
p = rep(1,12); dim(p) = c(2,3,2)
p[1,1,2] = 2; p[2,2,2]=2; p[,3,] = 0
TH = c(1:2)*0; dim(TH) = c(1,2)
res_d <- MRGVARData(m=2,n=2,p=p,TH=TH,T=100,S=2,SESVI=c(1,3),type="const")
max(res_d$Y)

### estimation of the MRGVAR model
colnames(res_d$Y) = c("P","Q","Pa","Qa")
res_e = MRGVARest(res=res_d)
res_e$Summary

IRF_CB  = irf_MRGVAR_CB(res=res_e,nstep=10,comb=NA,state=c(1,1),irf="gen1",runs=20,
conf=c(0.05,0.95))
IRF_g = IRF_graph(IRF_CB[[1]],Names=c("P","Q","Pa","Qa"))    #IRF
#IRF_g = IRF_graph(IRF_CB[[2]])   # accumulated IRF
```

`MRGVAR_Select` *Calculation of information criteria for a MRGVAR model*

Description

Calculation of information criteria for a MRGVAR model

Usage

```
MRGVAR_Select(res, I, L_V, TH_V)
```

Arguments

`res`	: a MRGVAR object obtained from MRGVARData or estimated from MRGVARest.
`I`	: index of the country under investigation.
`L_V`	: a four components vector containing the maxims of the domestic lag and the foreign lag for each regime, respectively.
`TH_V`	: a vector containing possible threshold values.

Value

a matrix with different lag specifications and the corresponding values of the model selection criteria.

Examples

```
p = c(2,2,2,2,2,2,2,2,2,2,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,0,0,0,0,0); dim(p) = c(5,3,2)
p[,2,] = 1; p[1,1,1] = 1; p[3,1,1] = 1; p[2,1,2] = 1

TH = c(1:5)*0; dim(TH) = c(1,5)
res_d <- MRGVARData(m=3,n=5,p=p,TH =TH,T=400,S=2,SESVI=((1:5)*3-2))
max(abs(res_d$Y)) # to make sure it is not explosive
colnames(res_d$Y) = c("Y11","Y21","Y31","Y12","Y22","Y32","Y13","Y23","Y33","Y14",
```

```
"Y24","Y34","Y15","Y25","Y35")
res_e = MRGVARest(res=res_d)

### four numbers for the maxima lag length in country I:
### regime 1: domestic foreign regime 2: domestic and foreign
L_v  = c(3,3,3,3)
### a vector containing possible threshold values
TH_v = c(-0.1, -0.05, 0,0.05, 0.1  )
CC = MRGVAR_Select(res=res_d,I=1,L_V=L_v,TH_V=TH_v)
 CCC = CC[[1]]

CCC[which.min(CCC[,9]),]
CCC[which.min(CCC[,18]),]

CC = MRGVAR_Select(res=res_d,I=2,L_V=L_v,TH_V=TH_v)
CCC = CC[[1]]

CCC[which.min(CCC[,9]),]
CCC[which.min(CCC[,18]),]
```

MRVARData *Data generating process of MRVAR(n,p,S)*

Description

This function generates data from an multi-regime stationary VAR(p) process and returns an MRVAR(n,p,S) object that is a list containing data and the parameters used in the MRVAR(n,p,S) process.

Usage

```
MRVARData(
  n,
  p,
  T,
  S,
  SESVI,
  TH,
  Bo,
  Co,
  Sigmao,
  Uo,
  SV,
  type,
  X,
  mu,
  Yo,
  Do,
  d
)
```

Arguments

n	: number of variables
p	: an (S x 2) matrix. Each row of p specifies the lag length and the number of exogenous variables for the corresponding regime.
T	: number of observations
S	: number of regimes
SESVI	: index of the switching variable
	`(n,S,p,T,SESVI) must be provided.`
TH	: an (S-1)-vector of threshold values
Bo	: an (n,n,p,S) array of the coefficients of the MRVAR(n,p,S) model. If Bo is not given it will be generated.
Co	: an (n,k+1,S) array of the coefficients of the deterministic components. For type="none" Co = O*(1:n,1:S), for "const" Co is an (1:n,1:S) array, for "exog0" Co is an (n,k+1,S) array with first column zero for each regime respectively,for "exog1" Co is an (n,1+k, S) array without zero restrictions.
Sigmao	: an (n,n,S) array containing S covariance matrices of residuals, each for one regime.
Uo	: residuals, if it is not NA it will be used as input to generate data of the MR-VAR(n,p,S) process.
SV	: exogenous switching variable
type	: type of the deterministic components type = c("none","const","exog0","exog1")
X	: a (T x k x S) array of exogenous variables for each state. The second dimension can be filled with zeros to take into account that the the exogenous variables are not identical in each state.
mu	: an (n x S) matrix of the regime specific means of the variables
Yo	: a (p, n, S) array of initial values of the process
Do	: a (T, n, S) array of extra exogenous components (not used with value zero)
d	: lag delays of the self-exiting switching variable. (TH,Bo,Co,Sigmao,Uo,SV,type) if not provided, they will be generated randomly.

Value

an MRVAR(n,p,S) object that is a list containing the generated data, the used parameters and the exogenous variables.

Examples

```
p = matrix(c(3,3,0,0),2,2)
res_d = MRVARData(n=2,p=p,T=200,S=2,SESVI=1,type="const")
colnames(res_d$Y) = c("Y","P")
STAT(res_d$B[,,,1])
STAT(res_d$B[,,,2])
res_e = MRVARest(res=res_d)
res_e$Summary
```

MRVARest *Estimation of MRVAR(n,p,S)*

Description

This function estimates the parameters of a specified MRVAR(n,p,S) based on provided data.

Usage

```
MRVARest(res)
```

Arguments

res	:an object of MRVAR that is an output of MRVARData including at least: n, S, p, type, Y, SESVI, TH, d, and optionally X.

Value

an MRVAR object that is a list containing estimated parameters and some test statistics.

Examples

```
p = matrix(c(3,3,0,0),2,2)
res_d = MRVARData(n=2,p=p,T=200,S=2,SESVI=1,type="const")
colnames(res_d$Y) = c("Y","P")
STAT(res_d$B[,,,1])
STAT(res_d$B[,,,2])
res_e = MRVARest(res=res_d)
res_e$Summary
```

MRVAR_Select *Calculation of information criteria for MRVAR(n,p,S) models.*

Description

Calculation of information criteria for MRVAR(n,p,S) models.

Usage

```
MRVAR_Select(res, L_V = L_V, TH_V = TH_V)
```

Arguments

res	: an MRVAR object generated from MRVARData or estimated from MRVARest.
L_V	: a 2 components vector containing the maxima lags in the two regimes, respectively.
TH_V	: a vector containing the possible threshold values over which the model selection criteria values will be calculated.

Value

a matrix with different lag specifications and threshold values as well as the corresponding information criterion values.

Examples

```
res_d = MRVARData(n=4,p=matrix(c(2,1,2,2,0,0,0,0),4,2),T=800,S=2,SESVI=1)
max(res_d$Y)
colnames(res_d$Y) = c("P","Y","R","U")
res_e = MRVARest(res=res_d)
TH_v = c(0,0.0)
L_v = c(5,5)
Sel = MRVAR_Select(res=res_e,L_V=L_v,TH_V=TH_v)
MRVAR_Select_Summary(Sel)
```

`MRVAR_Select_Summary` *Summary result of MRVAR model selection*

Description

Summary result of MRVAR model selection

Usage

```
MRVAR_Select_Summary(Sel)
```

Arguments

`Sel`	An output of MRVAR_Select

Value

The optimal model according to BIC or AIC

`STAT` *Root of the characteristic polynomial in Lag*

Description

Root of the characteristic polynomial in Lag

Usage

```
STAT(G)
```

Arguments

`G`	a coefficient matrix of dimension (n x n x p) of an VAR(p) object.

Value

Roots of the characteristic polynomial of VAR(p) `strsplit1` *Split a string*

Description

Split a string

Usage

```
strsplit1(x, split)
```

Arguments

`x`	A character vector with one element.
`split`	What to split on.

Value

A character vector.

Examples

```
x <- "alfa,bravo,charlie,delta"
strsplit1(x, split = ",")
```

`Type` *Check the consistency between type, Co, and EXOG*

Description

This function will output type according to specification of Co and EXOG and type.

Usage

```
Type(Co, EXOG, type)
```

Arguments

`Co`	The coefficients of the deterministic components
`EXOG`	The exogenous variables
`type`	The type of the deterministic specification for comparison

Value

type

Examples

```
Type(Co=matrix(c(0,0,1,1),2,2),EXOG= c(1:10),type="exog0")
Type(Co=matrix(c(1,1,1,1),2,2),EXOG= c(1:10),type="exog1")
Type(Co=matrix(c(1,1,1,1,0,0),2,3),EXOG= c(1:10),type=NA)
Type(Co=c(1,1),EXOG= NA,type=NA)
Type(Co=c(0,0),EXOG= NA,type=NA)
VARData(n=2,p=2,T=100,Co=matrix(c(2,2,1,1),2,2),type="exog0",X=c(1:100))
VARData(n=2,p=2,T=100,Co=matrix(c(1,1,1,1),2,2),type="exog1",X=c(1:100))
```

`VARData` *Data generating process of VAR(p)*

Description

This function will generate data from a stationary VAR(p) process and return a list containing data and parameters used in the VAR(p) process.

Usage

```
VARData(n, p, T, r_np, A, B, Co, U, Sigma, type, X, mu, Yo)
```

Arguments

`n`	: number of variables
`p`	: lag length
`T`	: number of observations
`r_np`	: an n x p matrix of roots of the characteristic polynomials of n independent AR(p)-processes. If not provided, it will be generated randomly.
`A`	: an n x n full rank matrix of transformation to generate correlated VAR(p) from the n independent AR(p)
`B`	: (n,n,p) array of the coefficients of the VAR(p) process. If B is not given, it will be calculated out of r_np and A.
`Co`	: (n,k+1) matrix of the coefficients of deterministic components in a VAR(p) process. For type="none" Co = 0*(1:n), for type="const" Co is an n vector, for type="exog0" Co is an (n,k) matrix, and for type="exog1" Co is an (n,1+k) matrix.
`U`	: residuals, if it is not NA it will be used as input to generate the VAR(p) process.
`Sigma`	: an n x n covariance matrix of the VAR(p) residuals.
`type`	: types of deterministic components. "none", "const" "exog0" and "exog1" are four options
`X`	: a (T x k) matrix of exogenous variables.
`mu`	: an n vector of the expected mean of the VAR(p) process
`Yo`	: a p x n matrix of initial values of the VAR(p) process

Value

An object of VAR(p) containing the generated data, the used parameters and the exogenous variables. res = list(n,p,type,r_np,Phi,A,B,Co,Sigma,Y,X,resid,U,Y1,Yo,check)

Examples

```
res_d = VARData(n=2,p=2,T=100,type="const")
res_d = VARData(n=2,p=2,T=10,Co=c(1:2)*0,type="none")
res_d = VARData(n=2,p=2,T=10,Co=c(1:2),  type="const")
res_d = VARData(n=3,p=2,T=200,type="exog1",X=matrix(rnorm(400),200,2))
res_d = VARData(n=3,p=2,T=200,Co=matrix(c(0,0,0,1,2,3,3,2,1),3,3), type="exog0",
X=matrix(rnorm(400),200,2))

res_d = VARData(n=2,p=2,T=100,type="const")
res_d = VARData(n=3,p=2,T=200,type="exog1",X=matrix(rnorm(400),200,2))
```

VARest *Estimation of VAR(p)*

Description

This function estimates parameters of a specified VAR(p) model based on provided data.

Usage

```
VARest(res)
```

Arguments

`res`	:a VAR(p) object which is an output of VARData including at least: n, p, type, Y and optionally X and type.

Value

res :a VAR(p) containing estimated parameter values, AIC, BIC and LH.

Examples

```
res_d = VARData(n=2,p=2,T=100,type="const")
res_e = VARest(res=res_d);
IRF_CB = irf_VAR_CB(res=res_e,nstep=20, comb=NA, irf = "gen1", runs = 100, conf = c(0.05, 0.95))
IRF_list <-IRF_graph(IRF_CB)
res_e$Summary
```

Index