Package ‘MTS’  
October 10, 2018

Type Package

Title All-Purpose Toolkit for Analyzing Multivariate Time Series (MTS)  
and Estimating Multivariate Volatility Models

Version 1.0

Date 2018-10-8

Author Ruey S. Tsay and David Wood

Maintainer Ruey S. Tsay <ruey.tsay@chicagobooth.edu>

Description Multivariate Time Series (MTS) is a general package for analyzing multivariate linear time series and estimating multivariate volatility models. It also handles factor models, constrained factor models, asymptotic principal component analysis commonly used in finance and econometrics, and principal volatility component analysis. (a) For the multivariate linear time series analysis, the package performs model specification, estimation, model checking, and prediction for many widely used models, including vector AR models, vector MA models, vector ARMA models, seasonal vector ARMA models, VAR models with exogenous variables, multivariate regression models with time series errors, augmented VAR models, and Error-correction VAR models for co-integrated time series. For model specification, the package performs structural specification to overcome the difficulties of identifiability of VARMA models. The methods used for structural specification include Kronecker indices and Scalar Component Models. (b) For multivariate volatility modeling, the MTS package handles several commonly used models, including multivariate exponentially weighted moving-average volatility, Cholesky decomposition volatility models, dynamic conditional correlation (DCC) models, copula-based volatility models, and low-dimensional BEKK models. The package also considers multiple tests for conditional heteroscedasticity, including rank-based statistics. (c) Finally, the MTS package also performs forecasting using diffusion index, transfer function analysis, Bayesian estimation of VAR models, and multivariate time series analysis with missing values. Users can also use the package to simulate VARMA models, to compute impulse response functions of a fitted VARMA model, and to calculate theoretical cross-covariance matrices of a given VARMA model.

License Artistic License 2.0

Imports Rcpp, fGarch, fBasics, mvtnorm

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-10-10 14:50:03 UTC
R topics documented:

MTS-package .................................................. 4
apca ................................................................. 4
archTest .......................................................... 5
backtest ........................................................... 6
BEKK11 ............................................................. 7
Btfm2 ............................................................... 8
BVAR ................................................................. 9
ccm ................................................................. 11
comVol ............................................................. 12
Corner .............................................................. 13
dccFit .............................................................. 13
dccPre .............................................................. 14
diffM ................................................................. 16
Eccm ................................................................. 16
ECMvar ............................................................. 17
ECMvar1 ........................................................... 19
EWMAvol .......................................................... 20
FEVdec .............................................................. 21
GrangerTest ....................................................... 22
hfactor .............................................................. 23
ibmspko ............................................................. 24
Kronfit .............................................................. 24
Kronid ............................................................... 26
Kronpred ........................................................... 27
Kronspec ........................................................... 28
MarchTest .......................................................... 29
MChdiag ............................................................ 31
MCholV ............................................................. 32
Mlm ................................................................. 33
mq ................................................................. 33
msqrt ............................................................... 34
mtCopula .......................................................... 35
MTS-internal ...................................................... 36
MTSdiag ........................................................... 36
MTSplot ............................................................ 37
Mtxprod ............................................................ 38
Mtxprod1 ........................................................... 38
Plwgt ............................................................... 39
PSIwgt ............................................................. 40
qgdps ............................................................... 41
refECMvar ........................................................ 41
refECMvar1 ........................................................ 42
refKronfit ........................................................ 43
refREGts .......................................................... 44
refSCMfit ........................................................ 44
refsVARMA ........................................................ 45
<table>
<thead>
<tr>
<th>R topics documented:</th>
</tr>
</thead>
<tbody>
<tr>
<td>refVAR</td>
</tr>
<tr>
<td>refVARMA</td>
</tr>
<tr>
<td>refVARX</td>
</tr>
<tr>
<td>refVMA</td>
</tr>
<tr>
<td>refVMAe</td>
</tr>
<tr>
<td>REGts</td>
</tr>
<tr>
<td>REGtspred</td>
</tr>
<tr>
<td>RLS</td>
</tr>
<tr>
<td>SCCor</td>
</tr>
<tr>
<td>SCMfit</td>
</tr>
<tr>
<td>SCMId</td>
</tr>
<tr>
<td>SCMId2</td>
</tr>
<tr>
<td>SCMmod</td>
</tr>
<tr>
<td>sVARMA</td>
</tr>
<tr>
<td>sVARMACpp</td>
</tr>
<tr>
<td>sVARMApred</td>
</tr>
<tr>
<td>SWfore</td>
</tr>
<tr>
<td>tenstocks</td>
</tr>
<tr>
<td>tfm</td>
</tr>
<tr>
<td>tfm1</td>
</tr>
<tr>
<td>tfm2</td>
</tr>
<tr>
<td>VAR</td>
</tr>
<tr>
<td>VARMA</td>
</tr>
<tr>
<td>VARMACov</td>
</tr>
<tr>
<td>VARMACpp</td>
</tr>
<tr>
<td>VARMAirf</td>
</tr>
<tr>
<td>VARMApred</td>
</tr>
<tr>
<td>VARAsim</td>
</tr>
<tr>
<td>VARorder</td>
</tr>
<tr>
<td>VARorderI</td>
</tr>
<tr>
<td>VARpred</td>
</tr>
<tr>
<td>VARpsi</td>
</tr>
<tr>
<td>VARS</td>
</tr>
<tr>
<td>VARX</td>
</tr>
<tr>
<td>VARXirf</td>
</tr>
<tr>
<td>VARXorder</td>
</tr>
<tr>
<td>VARXpred</td>
</tr>
<tr>
<td>Vech</td>
</tr>
<tr>
<td>VechM</td>
</tr>
<tr>
<td>VMA</td>
</tr>
<tr>
<td>VMACpp</td>
</tr>
<tr>
<td>VMAe</td>
</tr>
<tr>
<td>VMAorder</td>
</tr>
<tr>
<td>VMAs</td>
</tr>
<tr>
<td>Vmiss</td>
</tr>
<tr>
<td>Vpmiss</td>
</tr>
</tbody>
</table>

**Index** 96
**Description**

Multivariate Time Series (MTS) is a general package for analyzing multivariate linear time series and estimating multivariate volatility models. It also handles factor models, constrained factor models, asymptotic principal component analysis commonly used in finance and econometrics, and principal volatility component analysis. (a) For the multivariate linear time series analysis, the package performs model specification, estimation, model checking, and prediction for many widely used models, including vector AR models, vector MA models, vector ARMA models, seasonal vector ARMA models, VAR models with exogenous variables, multivariate regression models with time series errors, augmented VAR models, and Error-correction VAR models for co-integrated time series. For model specification, the package performs structural specification to overcome the difficulties of identifiability of VARMA models. The methods used for structural specification include Kronecker indices and Scalar Component Models. (b) For multivariate volatility modeling, the MTS package handles several commonly used models, including multivariate exponentially weighted moving-average volatility, Cholesky decomposition volatility models, dynamic conditional correlation (DCC) models, copula-based volatility models, and low-dimensional BEKK models. The package also considers multiple tests for conditional heteroscedasticity, including rank-based statistics. (c) Finally, the MTS package also performs forecasting using diffusion index, transfer function analysis, Bayesian estimation of VAR models, and multivariate time series analysis with missing values. Users can also use the package to simulate VARMA models, to compute impulse response functions of a fitted VARMA model, and to calculate theoretical cross-covariance matrices of a given VARMA model.

**Details**

<table>
<thead>
<tr>
<th>Package</th>
<th>MTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>License</td>
<td>Artistic License 2.0</td>
</tr>
</tbody>
</table>

**Author(s)**

Ruey S. Tsay and David Wood
**archTest**

**Description**
Perform asymptotic PCA for a data set. Typically for cases in which the number of variables is greater than the number of data points.

**Usage**
apca(da, m)

**Arguments**
da A T-by-k data set matrix, where T is the sample size and k is the dimension
m The number of common factors

**Details**
Perform the PCA analysis of interchanging the roles of variables and observations.

**Value**
sdev Square root of the eigenvalues
factors The common factors
loadings The loading matrix

**Author(s)**
Ruey S. Tsay

**References**

**Examples**

```r
tn=matrix(rnorm(1200),12,100)
sp100=apca(tn,3)
```

---

**archTest**

**ARCH test for univariate time series**

**Description**
Perform tests to check the conditional heteroscedasticity in a time series. The Ljung-Box statistics of squared series and a rank-based Ljung-Box test are used.

**Usage**
archTest(rt, lag = 10)
Arguments

rt
A scalar time series. If rt is a matrix, only the first column is used.

lag
The number of lags of ACF used in the Ljung-Box statistics. The default is 10.

Details

The Ljung-Box statistics based on the squared series are computed first. The rank series of the squared time series is then used to test the conditional heteroscedasticity.

Value

The Q-statistic and its p-value. Also, the rank-based Q statistic and its p-value.

Author(s)

Ruey Tsay

See Also

MarchTest

Examples

rt=rnorm(200)
archTest(rt)

backtest Backtesting of a scalar ARIMA model

Description

Perform out-of-sample prediction of a given ARIMA model and compute the summary statistics

Usage

backtest(m1, rt, orig, h, xre=NULL, fixed=NULL, inc.mean=TRUE, reest=1)

Arguments

m1
An output of the arima command for scalar time series

rt
The time series under consideration

orig
The starting forecast origin. It should be less than the length of the underlying time series

h
The forecast horizon. For a given h, it computes 1-step to h-step ahead forecasts

inc.mean
A logical switch. It is true if mean vector is estimated.
fixed  A vector of the length of the number of coefficients of the ARIMA model. It is used in R for parameter constraint.
xre  A matrix containing the exogeneous variables used in the ARIMA model
reest  A control variable used to re-fit the model in prediction. The program will re-estimate the model for every new reest observations. The default is 1. That is, re-estimate the model with every new data point.

Details
Perform estimation-prediction-reestimation in the forecasting subsample, and to compute the summary statistics

Value
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>origion</td>
<td>Forecast origin</td>
</tr>
<tr>
<td>error</td>
<td>forecast errors</td>
</tr>
<tr>
<td>forecasts</td>
<td>forecasts</td>
</tr>
<tr>
<td>rmse</td>
<td>Root mean squared forecast errors</td>
</tr>
<tr>
<td>mabso</td>
<td>Mean absolute forecast errors</td>
</tr>
<tr>
<td>reest</td>
<td>Return the reest value</td>
</tr>
</tbody>
</table>

Author(s)
Ruey S. Tsay

References
Back-Test of a Transfer Function Model with Two Input Variables

Description
Perform back-test of transfer function model with 2 input variable. For a specified tfm2 model and a given forecast origin, the command iterated between estimation and 1-step ahead prediction starting at the forecast origin until the (T-1)th observation, where T is the sample size.

Usage
Btfm2(y,x,x2=NULL,wt=NULL,ct=NULL,orderN=c(1,0,0),orderS=c(0,0,0),sea=12, order1=c(0,1,0),order2=c(0,-1,0),orig=(length(y)-1))

Arguments
y Data vector of dependent variable
x Data vector of the first input (or independent) variable
x2 Data vector of the second input variable if any
t Data vector of a given deterministic variable such as time trend, if any
wt Data vector of co-integrated series between input and output variables if any
orderN Order (p,d,q) of the regular ARMA part of the disturbance component
orderS Order (P,D,Q) of the seasonal ARMA part of the disturbance component
sea Seasonality, default is 12 for monthly data
Order (r, s, b) of the transfer function model of the first input variable, where r and s are the degrees of denominator and numerator polynomials and b is the delay.

Order (r2, s2, b2) of the transfer function model of the second input variable, where 2r and s2 are the degrees of denominator and numerator polynomials and b2 is the delay.

Forecast origin with default being T-1, where T is the sample size.

Details
Perform out-of-sample 1-step ahead prediction to evaluate a fitted tfm2 model.

Value
- **ferror**: 1-step ahead forecast errors, starting at the given forecast origin.
- **mse**: Out-of-sample mean squared forecast errors.
- **rmse**: Root mean squared forecast errors.
- **mae**: Out-of-sample mean absolute forecast errors.
- **nobf**: The number of 1-step ahead forecast errors computed.
- **rAR**: Regular AR coefficients.

Author(s)
Ruey S. Tsay

References

See Also
tfm2

BVAR

Bayesian Vector Autoregression

Description
Estimate a VAR(p) model using Bayesian approach, including the use of Minnesota prior.

Usage

BVAR(z, p=1, C, V0, n0=5, Phi0=NULL, include.mean=T)
Arguments

z  A matrix of vector time series, each column represents a series.
p  The AR order. Default is p=1.
C  The precision matrix of the coefficient matrix. With constant, the dimension of C is (kp+1)-by-(kp+1). The covariance matrix of the prior for the parameter vec(Beta) is Kronecker(Sigma_a,C-inverse).
V0  A k-by-k covariance matrix to be used as prior for the Sigma_a matrix
n0  The degrees of freedom used for prior of the Sigma_a matrix, the covariance matrix of the innovations. Default is n0=5.
Phi0  The prior mean for the parameters. Default is set to NULL, implying that the prior means are zero.
include.mean  A logical switch controls the constant term in the VAR model. Default is to include the constant term.

Details

for a given prior, the program provide the posterior estimates of a VAR(p) model.

Value

est  Posterior means of the parameters
Sigma  Residual covariance matrix

Author(s)

Ruey S. Tsay

References

Tsay (2014, Chapter 2).

Examples

data("mts-examples",package="MTS")
z=log(qgdp[,3:5])
zt=diff(z)*100
C=0.1*diag(rep(1,7))
V0=diag(rep(1,3))
BVAR(zt,p=2,C,V0)
Cross-Correlation Matrices

Description

Computes sample cross-correlation matrices of a multivariate time series, including simplified ccm matrix and p-value plot of Ljung-Box statistics.

Usage

ccm(x, lags = 12, level = FALSE, output = T)

Arguments

x            A matrix of vector time series, each column represents a series.
lags         The number of lags of CCM to be computed. Default is 12.
level        A logical switch. When level=T, numerical values of CCM is printed. Default is no printing of CCM.
output       A logical switch. If output=F, no output is given. Default is with output.

Details

The p-value of Ljung-Box statistics does not include any adjustment in degrees of freedom.

Value

ccm          Sample cross-correlation matrices
pvalue       p-values for each lag of CCM being a zero matrix

Author(s)

Ruey S. Tsay

References


Examples

xt=matrix(rnorm(1500),500,3)
ccm(xt)
ccm(xt,lag=20)
comVol

**Common Volatility**

**Description**

Compute the principal volatility components based on the residuals of a VAR(p) model.

**Usage**

```r
comVol(rtn, m = 10, p = 1, stand = FALSE)
```

**Arguments**

- `rtn`: A T-by-k data matrix of k-dimensional asset returns
- `m`: The number of lags used to compute generalized cross-Kurtosis matrix
- `p`: VAR order for the mean equation
- `stand`: A logical switch to standardize the returns

**Details**

Perform a VAR(p) fit, if any. Then, use the residual series to perform principal volatility component analysis. The ARCH test statistics are also computed for the sample principal components.

**Value**

- `residuals`: The residuals of a VAR(p) fit
- `values`: Eigenvalues of the principal volatility component analysis
- `vectors`: Eigenvectors of the principal volatility component analysis
- `M`: The transformation matrix

**Author(s)**

Ruey S. Tsay and Y.B. Hu

**References**

Tsay (2014, Chapter 7)

**Examples**

```r
data("mts-examples", package="MTS")
zt=diffM(log(qgdp[,3:5]))
m1=comVol(zt,p=2)
names(m1)
```
Corner

**Description**

For a given dependent variable and an input variable, the program computes the Corner table for specifying the order \((r,s,d)\) of a transfer function.

**Usage**

\[
\text{Corner}(y, x, Nrow=11, Ncol=7)
\]

**Arguments**

- \(y\): A pre-whitened dependent (or output) variable.
- \(x\): A pre-whitened independent (or input) variable. It should be a white noise series.
- \(Nrow\): The number of rows of the Corner table. Default is 11.
- \(Ncol\): The number of columns of the Corner table. Default is 7.

**Details**

For the pair of pre-whitened output and input variables, the program compute the Corner table and its simplified version for specifying the order of a transfer function.

**Value**

- \(\text{corner}\): The Corner table.

**Author(s)**

Ruey S. Tsay

dccFit

**Dynamic Cross-Correlation Model Fitting**

**Description**

Fits a DCC model using either multivariate Gaussian or multivariate Student-t innovations. Two types of DCC models are available. The first type is proposed by Engle and the other is by Tse and Tsui. Both models appear in the Journal of Business and Economic Statistics, 2002.

**Usage**

\[
\text{dccFit}(rt, type = "TseTsui", theta = c(0.90, 0.02),
    ub = c(0.95, 0.049999), lb = c(0.4, 0.00001),
    cond.dist = "std", df = 7, m = 0)
\]
Arguments

rt The T-by-k data matrix of k-dimensional standardized asset returns. Typically, they are the standardized residuals of the command dccPre.

type A logical switch to specify the type of DCC model. Type="TseTsui" for Tse and Tsui's DCC model. Type = "Engle" for Engle's DCC model. Default is Tse-Tsui model.

theta The initial parameter values for theta1 and theta2

ub Upper bound of parameters

lb Lower bound of parameters

cond.dist Conditional innovation distribution with std for multivariate Student-t innovations.

df degrees of freedom of the multivariate Student-t innovations.

m For Tse and Tsui method only, m denotes the number of returns used in local correlation matrix estimation

Value

estimates Parameter estimates

Hessian Hessian matrix of the estimates

rho.t Time-varying correlation matrices. Each row contains elements of a cross-correlation matrix.

Author(s)

Ruey S. Tsay

References


See Also

dccPre

dccPre  Preliminary Fitting of DCC Models

Description

This program fits marginal GARCH models to each component of a vector return series and returns the standardized return series for further analysis. The garchFit command of fGarch package is used.
Usage

dccPre(rtn, include.mean = T, p = 0, cond.dist = "norm")

Arguments

rtn
A T-by-k data matrix of k-dimensional asset returns

include.mean
A logical switch to include a mean vector. Default is to include the mean.

p
VAR order for the mean equation

cond.dist
The conditional distribution of the innovations. Default is Gaussian.

Details

The program uses fGarch package to estimate univariate GARCH model for each residual series after a VAR(p) fitting, if any.

Value

marVol
A matrix of the volatility series for each return series

sresi
Standardized residual series

est
Parameter estimates for each marginal volatility model

se.est
Standard errors for parameter estimates of marginal volatility models

Note

fGarch package is used

Author(s)

Ruey S. Tsay

References


See Also
dccFit
**diffM**

*Difference of multivariate time series*

**Description**

Performs the difference operation of a vector time series

**Usage**

```r
diffM(zt, d = 1)
```

**Arguments**

- `zt`: A vector time series (T by k, with sample size T and dimension k)
- `d`: Order of differencing. Default is d=1.

**Details**

When d = 1, the command is equivalent to `apply(zt, 2, diff)`

**Value**

The differenced time series

**Author(s)**

Ruey S Tsay

**Examples**

```r
data("mts-examples", package="MTS")
zt=log(gdp[,3:5])
xt=diffM(zt)
```

---

**Eccm**

*Extended Cross-Correlation Matrices*

**Description**

Compute the extended cross-correlation matrices and the associated two-way table of p-values of multivariate Ljung-Box statistics of a vector time series.

**Usage**

```r
Eccm(zt, maxp = 5, maxq = 6, include.mean = FALSE, rev = TRUE)
```
**Arguments**

- **zt**: Data matrix (T-by-k) of a vector time series, where T is the sample size and k is the dimension.
- **maxp**: Maximum AR order entertained. Default is 5.
- **maxq**: Maximum MA order entertained. Default is 6.
- **include.mean**: A logical switch controlling the mean vector in estimation. Default assumes zero mean.
- **rev**: A logical switch to control the cross-correlation matrices used to compute the multivariate Ljung-Box statistics. Traditional way is to compute test statistics from lag-1 to lag-m. If rev = TRUE, then the test statistics are compute from lag-(m-1) to lag-m, from lag-(m-2) to lag-m, etc.

**Value**

- **pEccm**: A two-way table of the p-values of extended cross-correlation matrices.
- **vEccm**: The sample extended cross-correlation matrices.
- **ARcoef**: AR coefficient matrices of iterated VAR fitting.

**Author(s)**

Ruey S. Tsay

**References**


**Examples**

```r
zt = matrix(rnorm(900), 300, 3)
m1 = Eccm(zt)
```

---

**Description**

Performs estimation of an Error-Correction VAR(p) model using the Quasi Maximum Likelihood Method.

**Usage**

```r
ECMvar(x, p, ibeta, include.const = FALSE, fixed = NULL,
       alpha = NULL, se.alpha = NULL, se.beta = NULL, phi = NULL, se.phi = NULL)
```
Arguments

- **x**: A T-by-k data matrix of a k-dimensional co-integrated VAR process
- **p**: VAR order
- **ibeta**: Initial estimate of the co-integrating matrix. The number of columns of ibeta is the number of co-integrating series
- **include.const**: A logical switch to include a constant term in the model. The default is no constant
- **fixed**: A logical matrix to set zero parameter constraints.
- **alpha**: Initial estimate of alpha, if any
- **se.alpha**: Initial estimate of the standard error of alpha, if any
- **se.beta**: Initial estimate of the standard error of beta, if any
- **phip**: Initial estimate of the VAR coefficients, if any
- **se.phip**: Initial estimate of the standard error of the VAR coefficients, if any

Value

- **data**: The vector time series
- **ncoint**: The number of co-integrating series
- **arorder**: VAR order
- **include.const**: Logical switch to include constant
- **alpha, se.alpha**: Estimates and their standard errors of the alpha matrix
- **beta, se.beta**: Estimates and their standard errors of the beta matrix
- **aic, bic**: Information criteria of the fitted model
- **residuals**: The residual series
- **Sigma**: Residual covariance matrix
- **phip, se.phip**: Estimates and their standard errors of VAR coefficients

Author(s)

Ruey S. Tsay

References

Tsay (2014, Chapter 5)

See Also

ECMvar1
**Examples**

```r
cphi=matrix(c(0.5,-0.25,-1.0,0.5),2,2); ctheta=matrix(c(0.2,-0.1,-0.4,0.2),2,2)
Sieg=diag(2)
mma=VARMAsim(300,alags=c(1),malags=c(1),phi=cphi,theta=ctheta,sigma=Sieg)
zeta=mma$series[,c(2,1)]
betamatrix(c(1,0.5),2,1)
m1=ECMvar1(zeta,ibeta=beta)
names(m1)
```

---

**Description**

Perform least-squares estimation of an ECM VAR(p) model with known co-integrating processes

**Usage**

```r
ECMvar1(x, p, wt, include.const = FALSE, fixed = NULL, output = TRUE)
```

**Arguments**

- `x` A T-by-k data matrix of a k-dimensional co-integrated VAR process
- `p` VAR order
- `wt` A T-by-m data matrix of m-dimensional co-integrated process
- `include.const` A logical switch to include a constant term. Default is no constant.
- `fixed` A logical matrix to set zero parameter constraints
- `output` A logical switch to control output

**Value**

- `data` The vector time series
- `wt` The co-integrated series
- `arorder` VAR order
- `include.const` Logical switch to include constant
- `coef` Parameter estimates
- `aic,bic` Information criteria of the fitted model
- `residuals` The residual series
- `Sigma` Residual covariance matrix

**Author(s)**

Ruey S. Tsay
References


See Also

ECMvar

Examples

phi=matrix(c(0.5,-0.25,-1.0,0.5),2,2); theta=matrix(c(0.2,-0.1,-0.4,0.2),2,2)
Sig=diag(2)
mm=VARMAsim(300,arlags=c(1),malags=c(1),phi=phi,theta=theta,sigma=Sig)
zt=mm$series
wt=0.5*zt[,1]+zt[,2]
m1=ECMvar1(zt,3,wt)
names(m1)

EWMAvol

Description

Use exponentially weighted moving-average method to compute the volatility matrix

Usage

EWMAvol(rtn, lambda = 0.96)

Arguments

rtn A T-by-k data matrix of k-dimensional asset returns, assuming the mean is zero
lambda Smoothing parameter. The default is 0.96. If lambda is negative, then the multivariate Gaussian likelihood is used to estimate the smoothing parameter.

Value

Sigma.t The volatility matrix with each row representing a volatility matrix
return The data
lambda The smoothing parameter lambda used

Author(s)

Ruey S. Tsay
FEVdec

References


Examples

data("mts-examples",package="MTS")
rtln=log(ibmspk[,2:4]+1)
m1=EWMAvol(rtn)

---

FEVdec

**Forecast Error Variance Decomposition**

Description

Computes the forecast error variance decomposition of a VARMA model

Usage

FEVdec(Phi, Theta, Sig, lag = 4)

Arguments

- **Phi**: VAR coefficient matrices in the form Phi=[Phi1, Phi2, ..., Phip], a k-by-kp matrix.
- **Theta**: VMA coefficient matrices in form form Theta=[Theta1, Theta2, ..., Thetaq], a k-by-kq matrix.
- **Sig**: The residual covariance matrix Sigma, a k-by-k positive definite matrix.
- **lag**: The number of lags of forecast errors variance to be computed. Default is 4.

Details

Use the psi-weight matrices to compute the forecast error covariance and use Cholesky decomposition to perform the decomposition

Value

- **irf**: Impulse response matrices
- **orthirf**: Orthogonal impulse response matrices
- **Omega**: Forecast error variance matrices
- **OmegaR**: Forecast error variance decomposition

Author(s)

Ruey S. Tsay
References

Tsay (2014, Chapter 3)

Examples

p1=matrix(c(0.2,-0.6,0.3,1,1),2,2)
theta1=matrix(c(-0.5,0,0,-0.6),2,2)
Sig=matrix(c(3,1,1,1),2,2)
m1=FEVdec(p1,theta1,Sig)
names(m1)

GrangerTest

Granger Causality Test

Description

Performs Granger causality test using a vector autoregressive model

Usage

GrangerTest(X,p=1,include.mean=T,locInput=c(1))

Arguments

x a T-by-p data matrix with T denoting sample size and p the number of variables
p vector AR order.
include.mean Indicator for including a constant in the model. Default is TRUE.
locInput Locators for the input variables in the data matrix. Default is the first column
being the input variable. Multiple inputs are allowed.

Details

Perform VAR(p) and constrained VAR(p) estimations to test the Granger causality. It uses likelihood ratio and asymptotic chi-square.

Value

data Original data matrix
cnst logical variable to include a constant in the model
order order of VAR model used
coeff Coefficient estimates
constraints Implied constraints of Granger causality
aic, bic, hq values of information criteria
residuals residual vector
secoef standard errors of coefficient estimates
Description

Performs factor model analysis with a given constrained matrix

Usage

hfactor(x, H, r)

Arguments

x A T-by-k data matrix of an observed k-dimensional time series
H The constrained matrix with each column representing a constraint
r The number of common factor

Value

Results of the traditional PCA and constrained factor models are given

Author(s)

Ruey S. Tsay

References

Tsay (2014, Chapter 6). Tsai and Tsay (2010, JASA)
Examples

```r
data("mts-examples", package="MTS")
rt= log(tenstocks[,2:11]+1)  # compute log returns
h1=c(1,1,1,1,rep(0,6))    # specify the constraints
h2=c(0,0,0,0,1,1,0,0,0)
h3=c(rep(0.7),1,1,1)
H=cbind(h1,h2,h3)
m1=hfactor(rtn,H,3)
```

ibmspko

*Monthly simple returns of the stocks of International Business Machines (IBM) and Coca Cola (KO) and the S&P Composite index (SP)*

Description

Monthly simple returns of the stocks of International Business Machines (IBM) and Coca Cola (KO) and the S&P Composite index (SP). The sample period is from January 1961 to December 2011. The original data were from the Center for Research in Security Prices (CRSP) of the University of Chicago. The file has four columns. They are dates, IBM, SP, and KO.

Usage

ibmspko

Format

A 2-d list containing 612x4 observations. The file has four columns. They are dates, IBM, SP, and KO.

Source


Kronfit

*Fitting a VARMA Model via Kronecker Index*

Description

Perform estimation of a VARMA model specified by the Kronecker indices.

Usage

Kronfit(da, kidx, include.mean = T, fixed = NULL, Kpar=NULL, seKpar=NULL, prelim = F, details = F, thres = 1)
Arguments

da: Data matrix (T-by-k) of a k-dimensional time series
ki: The vector consisting of Kronecker indices
include.mean: A logical switch for including the mean vector in estimation. Default is to include the mean vector.
fixed: A logical matrix used to set zero parameter constraints. This is used mainly in the command refKronfit.
Kpar: Parameter vectors for use in model simplification
seKpar: Standard errors of the parameter estimates for use in model simplification
prelim: A logical switch for a preliminary estimation.
details: A logical switch to control output.
ths: A threshold for t-ratios in setting parameter to zero. Default is 1.

Value

data: The observed time series data
Kindex: Kronecker indices
ARid: Specification of AR parameters: 0 denotes fixing to zero, 1 denotes fixing to 1, and 2 denoting estimation
MAid: Specification of MA parameters
cnst: A logical variable: include.mean
coef: Parameter estimates
se.coef: Standard errors of the estimates
residuals: Residual series
sigma: Residual covariance matrix
aic,bic: Information criteria of the fitted model
ph: Constant vector
Phi: AR coefficient matrices
Theta: MA coefficient matrices

Author(s)
Ruey S. Tsay

References

See Also
refKronfit, Kronspec
Kronecker Index Identification

Description

Find the Kronecker indices of a k-dimensional time series

Usage

Kronid(x, plag = 5, crit = 0.05)

Arguments

x 
Data matrix (T-by-k) of a k-dimensional time series
plag 
The number of lags used to represent the past vector. Default is 5.
crit 
Type-I error used in testing for zero canonical correlations. Default is 0.05.

Value

index 
Kronecker indices
tests 
Chi-square test statistics

Author(s)

Ruey S. Tsay

References


Examples

phi=matrix(c(0.2,-0.6,.3,1.1),2,2); sigma=diag(2); theta=-0.5*sigma
ml=VARMAsim(300,arlags=c(1),malags=c(1),phi=phi,theta=theta,sigma=sigma)
zt=ml$series
Kronid(zt)
Kronpred

*Prediction of a fitted VARMA model via Kronfit, using Kronecker indices*

**Description**

Compute forecasts of a fitted VARMA model via the command Kronfit

**Usage**

```r
kronpred(model, orig=0, h=1)
```

**Arguments**

- `model`: A model fitted by the Kronfit command
- `orig`: Forecast origin. The default is 0, implying that the origin is the last observation
- `h`: Forecast horizon. Default is `h=1`, 1-step ahead forecast

**Details**

For a model, which is the output of the command Kronfit, the command computes forecasts of the model starting at the forecast origin. 1-step to h-step ahead forecasts are computed.

**Value**

- `pred`: Forecasts
- `se.err`: Standard errors of the forecasts
- `orig`: Return the forecast origin

**Author(s)**

Ruey S. Tsay

**References**

Kronspec

*Kronecker Index Specification*

**Description**

For a given set of Kronecker indices, the program specifies a VARMA model. It gives details of parameter specification.

**Usage**

\[
\text{Kronspec}(kdx, \text{output} = \text{TRUE})
\]

**Arguments**

- **kdx**
  A vector of Kronecker indices
- **output**
  A logical switch to control output. Default is with output.

**Value**

- **PhiID**
  Specification of the AR matrix polynomial. 0 denotes zero parameter, 1 denotes fixing parameter to 1, and 2 denotes the parameter requires estimation
- **ThetaID**
  Specification of the MA matrix polynomial

**Author(s)**

Ruey S. Tsay

**References**

Tsay (2014, Chapter 4)

**Examples**

```r
kdx=c(2,1,1)
m1=Kronspec(kdx)
names(m1)
```
MarchTest

Multivariate ARCH test

Description
Perform tests to check the conditional heteroscedasticity in a vector time series

Usage
MarchTest(zt, lag = 10)

Arguments
zt      a nT-by-k data matrix of a k-dimensional financial time series, each column contains a series.
lag     The number of lags of cross-correlation matrices used in the tests

Details
Several tests are used. First, the vector series zt is transformed into rt = t(zt) perform the test. The second test is based on the ranks of the transformed rt series. The third test is the multivariate Ljung-Box statistics for the squared vector series zt^2. The fourth test is the multivariate Ljung-Box statistics applied to the 5-percent trimmed series of the transformed series rt.

Value
Various test statistics and their p-values

Author(s)
Ruey S. Tsay

References

Examples
zt=matrix(rnorm(600),200,3)
MarchTest(zt)
function (zt, lag = 10)
{
  if (!is.matrix(zt))
    zt = as.matrix(zt)
  nT = dim(zt)[1]
  k = dim(zt)[2]
  C0 = cov(zt)
  zt1 = scale(zt, center = TRUE, scale = FALSE)
C0iv = solve(C0)
wk = zt1 \* C0iv
wk = wk \* zt1
rt2 = apply(wk, 1, sum) - k
m1 = acf(rt2, lag.max = lag, plot = F)
acf = m1%acf[2:(lag + 1)]
c1 = c(1:lag)
deno = rep(nT, lag) - c1
Q = sum(acf^2/deno) \* nT \* (nT + 2)
pv1 = 1 - pchisq(Q, lag)
cat("Q(m) of squared series (LM test): ", "\n")
cat("Test statistic: ", Q, " p-value: ", pv1, "\n")
rk = rank(rt2)
m2 = acf(rk, lag.max = lag, plot = F)
acf = m2%acf[2:(lag + 1)]
mu = -(rep(nT, lag) - c(1:lag))/(nT \* (nT - 1))
v1 = rep(5 \* nT^4, lag) - (5 \* c(1:lag) + 9) \* nT^3 + 9 \*
(c(1:lag) - 2) \* nT^2 + 2 \* c(1:lag) \* (5 \* c(1:lag) +
8) \* nT + 16 \* c(1:lag)^2
v1 = v1/(5 \* (nT - 1)^2 \* nT^2 \* (nT + 1))
QR = sum((acf - mu)^2/v1)
pv2 = 1 - pchisq(QR, lag)
cat("Rank-based Test: ", "\n")
cat("Test statistic: ", QR, " p-value: ", pv2, "\n")
cat("Q_k(m) of squared series: ", "\n")
x = zt^2
g0 = var(x)
ginv = solve(g0)
qm = 0
df = 0
for (i in 1:lag) {
x1 = x[(i + 1):nT, ]
x2 = x[1:(nT - i), ]
g = cov(x1, x2)
g = g \* (nT - i - 1)/(nT - 1)
h = t(g) \%x% ginv \%x% g \%x% ginv
qm = qm + nT \* nT \* sum(diag(h))/(nT - i)
df = df + k \* k
}
pv3 = 1 - pchisq(qm, df)
cat("Test statistic: ", qm, " p-value: ", pv3, "\n")
cut1 = quantile(rt2, 0.95)
idx = c(1:nT)[rt2 <= cut1]
x = zt[idx, ]^2
eT = length(idx)
g0 = var(x)
ginv = solve(g0)
qm = 0
df = 0
for (i in 1:lag) {
x1 = x[(i + 1):eT, ]
x2 = x[1:(eT - i), ]
g = cov(x1, x2)
Description

Apply four portmanteau test statistics to check the validity of a fitted multivariate volatility model.

Usage

`MCHdiag(at, Sigma.t, m = 10)`

Arguments

- **at**: A T-by-k matrix of residuals for a k-dimensional asset return series.
- **Sigma.t**: The fitted volatility matrices. The dimension is T-by-k^2 matrix.
- **m**: The number of lags used in the tests. Default is 10.

Details

The four test statistics are given in Tsay (2014, Chapter 7).

Value

Four test statistics and their p-values.

Author(s)

Ruey S. Tsay

References

Description

Use Cholesky decomposition to obtain multivariate volatility models

Usage

\[ \text{MCholV}(\text{rtn}, \text{size} = 36, \text{lambda} = 0.96, p = 0) \]

Arguments

- \textit{rtn} A T-by-k data matrix of a k-dimensional asset return series.
- \textit{size} The initial sample size used to start recursive least squares estimation
- \textit{lambda} The exponential smoothing parameter. Default is 0.96.
- \textit{p} VAR order for the mean equation. Default is 0.

Details

Use recursive least squares to perform the time-varying Cholesky decomposition. The least squares estimates are then smoothed via the exponentially weighted moving-average method with decaying rate 0.96. University GARCH(1,1) model is used for the innovations of each linear regression.

Value

- \textit{betat} Recursive least squares estimates of the linear transformations in Cholesky decomposition
- \textit{bt} The transformation residual series
- \textit{Vol} The volatility series of individual innovations
- \textit{Sigma.t} Volatility matrices

Author(s)

Ruey S. Tsay

References

Tsay (2014, Chapter 7)

See Also

fGarch
Fit a multivariate multiple linear regression model via the least squares method

Usage

\[
\text{Mlm}(y, z, \text{constant}=\text{TRUE}, \text{output}=\text{TRUE})
\]

Arguments

- \(y\) data matrix of dependent variable. Each column contains one variable.
- \(z\) data matrix of the explanatory variables. Each column contains one variable.
- \(\text{constant}\) A logical switch for including the constant term
- \(\text{output}\) A logical switch to print the output

Value

- \(\beta\) coefficient matrix
- \(\text{se.beta}\) standard errors of the coefficient matrix
- \(\text{residuals}\) The residual series
- \(\sigma\) Residual covariance matrix

Author(s)

Ruey S. Tsay

Computes the multivariate Ljung-Box statistics for cross-correlation matrices

Usage

\[
\text{mq}(x, \text{lag} = 24, \text{adj} = 0)
\]
Arguments

\( x \)  
The data matrix of a vector time series or residual series of a fitted multivariate model.

\( \text{lag} \)  
The number of cross-correlation matrices used. Default is 24.

\( \text{adj} \)  
Adjustment for the degrees of freedom for the Ljung-Box statistics. This is used for residual series. Default is zero.

Details

Computes the multivariate Ljung-Box statistics and their p-values. For model checking, the subcommand adj can be used to adjust the degrees of freedom of the Chi-square statistics.

Value

The multivariate Q-statistics and their p-values. Also, it provides a plot of the p-values.

Author(s)

Ruey S. Tsay

References


Examples

\[ x = \text{matrix}(\text{rnorm}(1500), 500, 3) \]
\[ \text{mq}(x) \]

---

\textit{msqrt}  
\textit{Square Root Matrix}

Description

Compute the symmetric square root of a positive definite matrix

Usage

\texttt{msqrt(M)}

Arguments

\( M \)  
A positive definite matrix

Details

Use spectral decomposition to compute the square root of a positive definite matrix
Value

- mtxsqrt: The square root matrix
- invsqrt: The inverse of the square root matrix

Note

This command is used in some of the MTS functions.

Author(s)

Ruey S. Tsay

Examples

```r
m = matrix(c(1, 0.2, 0.2, 1), 2, 2)
m1 = mtxsqrt(m)
names(m1)
```

Description

Fits a t-copula to a k-dimensional standardized return series. The correlation matrices are parameterized by angles and the angles evolve over time via a DCC-type equation.

Usage

```r
mtCopula(rt, g1, g2, grp = NULL, th0 = NULL, m = 0,
          include.th0 = TRUE, ub = c(0.95, 0.049999))
```

Arguments

- `rt`: A T-by-k data matrix of k standardized time series (after univariate volatility modeling)
- `g1`: lambda1 parameter, nonnegative and less than 1
- `g2`: lambda2 parameter, nonnegative and satisfying lambda1 + lambda2 < 1.
- `grp`: A vector to indicate the number of assets divided into groups. Default means each individual asset forms a group.
- `th0`: Initial estimate of theta0
- `m`: Number of lags used to estimate the local theta-angles
- `include.th0`: A logical switch to include theta0 in estimation. Default is to include.
- `ub`: Upper bound of parameters
Value

- estimates: Parameter estimates
- Hessian: Hessian matrix
- rho.t: Cross-correlation matrices
- theta.t: Time-varying angle matrices

Author(s)

Ruey S. Tsay

References


### MTS-internal

**MTS Internal Functions**

#### Description

MTS Internal Functions

#### Details

These are not to be called by the user.

### MTSdiag

**Multivariate Time Series Diagnostic Checking**

#### Description

Performs model checking for a fitted multivariate time series model, including residual cross-correlation matrices, multivariate Ljung-Box tests for residuals, and residual plots

#### Usage

```r
MTSdiag(model, gof = 24, adj = 0, level = F)
```

#### Arguments

- **model**: A fitted multivariate time series model
- **gof**: The number of lags of residual cross-correlation matrices used in the tests
- **adj**: The adjustment for degrees of freedom of Ljung-Box statistics. Typically, the number of fitted coefficients of the model. Default is zero.
- **level**: Logical switch for printing residual cross-correlation matrices
Value
Various test statistics, their p-values, and residual plots.

Author(s)
Ruey S. Tsay

Examples
\[
\begin{align*}
\text{phi} &= \text{matrix}(c(0.2, -0.6, 0.3, 1.1), 2, 2); \text{ sigma} = \text{diag}(2) \\
\text{m1} &= \text{VAR\text{Asim}(200, ar\text{lags}=c(1), phi=phi, sigma=sigma)} \\
\text{zt} &= \text{m1}\$\text{series} \\
\text{m2} &= \text{VAR}(\text{zt}, 1, \text{include.mean}=\text{FALSE}) \\
\text{MTS\text{diag}(m2)}
\end{align*}
\]

---

\textbf{Description}
Provides time plots of a vector time series

\textbf{Usage}
\begin{verbatim}
MTS\text{plot}(\text{data, caltime = NULL})
\end{verbatim}

\textbf{Arguments}
\begin{itemize}
\item \textbf{data} \quad \text{data matrix of a vector time series}
\item \textbf{caltime} \quad \text{Calendar time. Default is NULL, that is, using time index}
\end{itemize}

\textbf{Details}
Provides time plots of a vector time series. The output frame depends on the dimension of the time series

\textbf{Value}
Time plots of vector time series

\textbf{Author(s)}
Ruey S. Tsay

\textbf{Examples}
\begin{verbatim}
x\text{t} &= \text{matrix}(\text{rnorm(1500)}, 500, 3) \\
\text{MTS\text{plot}(xt)}
\end{verbatim}
**Mtxprod**  
*Polynomial Matrix Product*

**Description**
Compute the product of two polynomial matrices

**Usage**
```
Mtxprod(Mtx, sMtx, p, P)
```

**Arguments**
- **Mtx**: The coefficient matrix of a regular polynomial matrix
- **sMtx**: The coefficient matrix of a seasonal polynomial matrix
- **p**: Degree of the regular polynomial matrix
- **P**: Degree of the seasonal polynomial matrix

**Value**
Coefficient matrix of the product. The product is in the form \( \text{reg-AR} \cdot s\text{AR} \), etc.

**Author(s)**
Ruey S. Tsay

---

**Mtxprod1**  
*Alternative Polynomial Matrix Product*

**Description**
Compute the product of two polynomial matrices

**Usage**
```
Mtxprod1(Mtx, sMtx, p, P)
```

**Arguments**
- **Mtx**: The coefficient matrix of a regular polynomial matrix
- **sMtx**: The coefficient matrix of a seasonal polynomial matrix
- **p**: Degree of the regular polynomial matrix. \( p \) is less than \( P \).
- **P**: Degree of the seasonal polynomial matrix
Details

This polynomial product is used in seasonal VARMA modeling to check the multiplicative nature between the regular and seasonal polynomial matrices.

Value

Coefficient matrix of the product. The product matrix is in the form sAR * reg-AR, etc.

Author(s)

Ruey S. Tsay

---

**Pi Weight Matrices**

Description

Compute the Pi-weight matrices of a VARMA model.

Usage

`PIwgt(Phi = NULL, Theta = NULL, lag = 12, plot = TRUE)`

Arguments

- **Phi**: A k-by-kp matrix of VAR coefficients in the form [Phi1, Phi2, Phi3, ..., Phip]
- **Theta**: A k-by-kq matrix of VMA coefficients in the form [Theta1, Theta2, ..., Thetaq]
- **lag**: The number of Pi-weight matrices to be computed.
- **plot**: A logical switch to plot the Pi-weight matrices.

Details

The Pi-weight matrices for a VARMA model is \( \Pi(B) = \text{inverse}(Theta(B)) \times \Phi(B) \).

Value

- **pi.weight**: The matrix of Pi-weight coefficient.

Author(s)

Ruey S. Tsay

References

See Also

PSIwgt

Examples

phi = matrix(c(0.2, -0.6, 0.3, 1.1), 2, 2)
theta = matrix(c(-0.5, 0.2, 0.0, -0.6), 2, 2)
m1 = PSIwgt(Phi = phi, Theta = theta)
**qgdp**

Quarterly real gross domestic products of United Kingdom, Canada, and the United States

**Description**

Quarterly real gross domestic products of United Kingdom, Canada, and the United States from the first quarter of 1980 to the second quarter of 2011. The UK and CA data were originally from OECD and the US data from the Federal Reserve Bank of St Louis.

**Usage**

qgdp

**Format**

A 2-d list containing 126x5 observations. The data set consists of 5 columns: name, year, month, UK, CA, and US.

**Source**

The data were downloaded from the FRED of the Federal Reserve Bank of St Louis. The UK data were in millions of chained 2006 Pounds, the CA data were in millions of chained 2002 Canadian dollars, and the US data were in millions of chained 2005 dollars.

---

**refECMvar**

Refining Error-Correction Model for VAR series

**Description**

Refining an estimated ECM VAR model by setting insignificant estimates to zero

**Usage**

refECMvar(m1, thres = 1)

**Arguments**

- m1: An object of the ECMvar command or the refECMvar command
- thres: Threshold for individual t-ratio. The default is 1.

**Details**

Set simultaneously all estimates with t-ratio less than the threshold to zero (in modulus).
Value
Constrained estimation results of a ECM VAR model

Author(s)
Ruey S. Tsay

References

refECMvar1  Refining ECM for a VAR process

Description
Performs constrained least squares estimation of a ECM VAR model with known co-integrated processes

Usage
refECMvar1(m1, thres = 1)

Arguments
m1 An object of the ECMvar1 command or the refECMvar1 command
thres Threshold for individual t-ratio. Default is 1.

Details
Setting all estimates with t-ratio less than the threshold, in absolute value, to zero simultaneously.

Value
Constrained estimation results of an ECM VAR model

Author(s)
Ruey S. Tsay

References

See Also
ECMvar1, refECMvar
Description

This program performs model simplification of a fitted VARMA model via the Kronecker index approach.

Usage

refKronfit(model, thres = 1)

Arguments

- model: The name of a model from the command Kronfit or refKronfit
- thres: A threshold for t-ratio of individual parameter estimate. The default is 1.

Details

For a given threshold, the program sets a parameter to zero if its t-ratio (in modulus) is less than the threshold.

Value

Same as those of the command Kronfit.

Author(s)

Ruey S. Tsay

References


See Also

Kronfit
**refSCMfit**

*Refining Estimation of VARMA Model via SCM Approach*

**Description**
Refine estimation of a VARMA model specified via the SCM approach by removing insignificant parameters

**Usage**
refSCMfit(model, thres = 1)

**Arguments**
- model: An output object from the VARMA command or refVARMA command
- thres: Threshold value for individual t-ratio. Default is 1.

**Value**
The same as those of the command VARMA.

**Author(s)**
Ruey S. Tsay

**References**

**See Also**
refVAR, refVARMA

---

**refREGts**

*Refining a Regression Model with Time Series Errors*

**Description**
Refines a fitted REGts by setting simultaneously parameters with t-ratios less than the threshold (in modulus) to zero

**Usage**
refREGts(m, thres = 1)

**Arguments**
- m: An output object from the REGts command or refREGts command
- thres: Threshold value for individual t-ratio. Default is 1.

**Value**
The same as those of the command REGts.

**Author(s)**
Ruey S. Tsay

**References**

**See Also**
refV AR, refV ARMA
**refsVARMA**

**Arguments**

- `model` Name of the model from the SCMfit command or the refSCMfit command
- `thres` Threshold for the t-ratio of individual coefficient. Default is 1.

**Value**

The same as those of the command SCMfit.

**Author(s)**

Ruey S. Tsay

**References**

Tsay (2014, Chapter 4)

**See Also**

SCMfit

---

**refsVARMA**  
*Refining a Seasonal VARMA Model*

**Description**

Refines a fitted seasonal VARMA model by setting insignificant estimates to zero

**Usage**

```r
refsVARMA(model, thres = 0.8)
```

**Arguments**

- `model` An output object of the sVARMA command or the refsVARMA command
- `thres` Threshold for individual t-ratio. Default is 0.8.

**Details**

The command removes simultaneously all parameters with t-ratio less than the threshold in modulus.

**Value**

The same as those of the command sVARMA

**Author(s)**

Ruey S. Tsay
References

Tsay (2014, Chapter 6)

See Also

sVARMA

refVAR Refining a VAR Model

Description

Refine a fitted VAR model by removing simultaneously insignificant parameters

Usage

refVAR(model, fixed = NULL, thres = 1)

Arguments

model An output object of the command VAR or the refVAR command
fixed A logical matrix for VAR polynomial structure
thres Threshold used to set parameter to zero. Default is 1.

Details

Refine a VAR fitting by setting all estimates with t-ratio less than the threshold (in modulus) to zero.

Value

The same as those of the command VAR

Author(s)

Ruey S. Tsay

References

Tsay (2014, Chapter 2)

See Also

VAR


**Examples**

```r
data("mts-examples", package="MTS")
gdp=log(qgdp[,3:5])
zeta=diffM(gdp)
m1=VAR(zeta,3)
m2=refVAR(m1, thres=1.0)
names(m2)
```

---

**Description**

Refines a fitted VARMA model by setting insignificant estimates to zero.

**Usage**

```r
refVARMA(model, thres = 1.5)
```

**Arguments**

- `model`: An output object from the command VARMA or the command refVARMA.
- `thres`: A threshold value for individual t-ratio of the estimates.

**Details**

The program simultaneously sets estimates with t-ratios less than the threshold (in modulus) to zero.

**Value**

The same as those of the command VARMA.

**Author(s)**

Ruey S. Tsay

**References**


**See Also**

VARMA
refVARX  

Refining a VARX Model

Description
Refine a fitted VARX model by setting insignificant parameters to zero.

Usage
refVARX(m1, thres = 1)

Arguments
- m1: An output object of the VARX command or the refVARX command.
- thres: A threshold for the individual t-ratio. Default is 1.

Details
The program sets simultaneously all estimates with t-ratio less than threshold (in modulus) to zero and re-estimate the VARX model.

Value
The same as those of the command VARX.

Author(s)
Ruey S. Tsay

References

See Also
VARX
Refining VMA Models

Description

Refines a fitted VMA model by setting insignificant parameters to zero.

Usage

`refVMA(model, thres = 1)`

Arguments

- `model`: An output object from the command `VMA` or the `refVMA` command.
- `thres`: A threshold for individual t-ratio of parameter estimate. Default is 1.

Details

The program simultaneously sets all estimates with t-ratios less than the threshold (in modulus) to zero.

Value

The same as those of the command `VMA`.

Author(s)

Ruey S. Tsay

References


See Also

VMA
Refining VMA Estimation via the Exact Likelihood Method

Description

Refines a fitted VMA model via the VMAe command by setting insignificant parameters to zero

Usage

refVMAe(model, thres = 1)

Arguments

model An output object of the command VMAe or the command refVMAe itself
thres A threshold for individual t-ratio of parameter estimates. Default is 1.

Details

The program sets simultaneously all estimates with t-ratios less than the threshold (in modulus) to zero.

Value

The same as those of the command VMAe.

Author(s)

Ruey S. Tsay

References


See Also

VMAe, refVMA
**Regression Model with Time Series Errors**

**Description**

Perform the maximum likelihood estimation of a multivariate linear regression model with time-series errors

**Usage**

`REGts(zt, p, xt, include.mean = T, fixed = NULL, par = NULL, se.par = NULL, details = F)`

**Arguments**

- `zt` A T-by-k data matrix of a k-dimensional time series
- `p` The VAR order
- `xt` A T-by-v data matrix of independent variables, where v denotes the number of independent variables (excluding constant 1).
- `include.mean` A logical switch to include the constant term. Default is to include the constant term.
- `fixed` A logical matrix used to set parameters to zero
- `par` Initial parameter estimates of the beta coefficients, if any.
- `se.par` Standard errors of the parameters in par, if any.
- `details` A logical switch to control the output

**Details**

Perform the maximum likelihood estimation of a multivariate linear regression model with time series errors. Use multivariate linear regression to obtain initial estimates of regression coefficients if not provided

**Value**

- `data` The observed k-dimensional time series
- `xt` The data matrix of independent variables
- `aror` VAR order
- `include.mean` Logical switch for the constant vector
- `Phi` The VAR coefficients
- `se.Ph` The standard errors of Phi coefficients
- `beta` The regression coefficients
- `se.beta` The standard errors of beta
- `residuals` The residual series
- `sigma` Residual covariance matrix
- `coef` Parameter estimates, to be used in model simplification.
- `se.coef` Standard errors of parameter estimates
**Author(s)**
Ruey S. Tsay

**References**

---

**DESCRIPTION**

`regtspred` is a function for predicting a fitted regression model with time series errors.

**Usage**

```r
regtspred(model, newxt, h=1, orig=0)
```

**Arguments**

- `model`: An output of the REGts command for a vector time series with exogenous variables.
- `newxt`: The new data matrix of the exogenous variables. It must be of the same dimension as the original exogenous variables and of length at least `h` (the forecast horizon).
- `orig`: The forecast origin. The default is zero indicating that the origin is the last observation.
- `h`: The forecast horizon. For a given `h`, it computes 1-step to `h`-step ahead forecasts. Default is 1.

**Details**

Perform prediction of a fitted REGts model

**Value**

- `pred`: Forecasts
- `se.err`: Standard errors of forecasts
- `rmse`: Root mean squares of forecast errors
- `rmse`: Root mean squared forecast errors
- `orig`: Return the forecast origin

**Author(s)**
Ruey S. Tsay
References


---

**RLS**

*Recursive Least Squares*

**Description**

Compute recursive least squares estimation

**Usage**

```r
RLS(y, x, ist = 30, xpi = NULL, xpy0 = NULL)
```

**Arguments**

- `y`: data of dependent variable
- `x`: data matrix of regressors
- `ist`: initial number of data points used to start the estimation
- `xpi`: inverse of the X’X matrix
- `xpy0`: initial value of X’y.

**Value**

- `beta`: time-varying regression coefficient estimates
- `resi`: the residual series of recursive least squares estimation

**Note**

This function is used internally, but can also be used as a command.

**Author(s)**

Ruey S. Tsay
**SCCor**

*Sample Constrained Correlations*

**Description**

Compute the sample constrained correlation matrices

**Usage**

```r
SCCor(rt, end, span, grp)
```

**Arguments**

- `rt`: A T-by-k data matrix of a k-dimensional time series
- `end`: The time index of the last data point to be used in computing the sample correlations.
- `span`: The size of the data span used to compute the correlations.
- `grp`: A vector of group sizes. The time series in the same group are pooled to compute the correlation matrix.

**Value**

- `unconCor`: Un-constrained sample correlation matrix
- `conCor`: Constrained sample correlation matrix

**Note**

This is an internal function, not intended to be a general command

**Author(s)**

Ruey S. Tsay

**Examples**

```r
r=matrix(rnorm(1000),200,5)
grp=c(3,2)
m1=SCCor(r,200,200,grp)
m1$unconCor
m1$conCor
```
SCMfit

Scalar Component Model Fitting

Description

Perform estimation of a VARMA model specified via the SCM approach.

Usage

```r
SCMfit(da, scms, Tdx, include.mean = T, fixed = NULL,
       prelim = F, details = F, thres = 1, ref = 0,
       SCMpar=NULL, seSCMpar=NULL)
```

Arguments

- `da`: The T-by-k data matrix of a k-dimensional time series
- `scms`: A k-by-2 matrix of the orders of SCMs
- `Tdx`: A k-dimensional vector for locating "1" of each row in the transformation matrix.
- `include.mean`: A logical switch to include the mean vector. Default is to include mean vector.
- `fixed`: A logical matrix to set parameters to zero
- `prelim`: A logical switch for preliminary estimation. Default is false.
- `details`: A logical switch to control details of output
- `thres`: Threshold for individual t-ratio when setting parameters to zero. Default is 1.
- `ref`: A switch to use SCMmod in model specification.
- `SCMpar`: Parameter estimates of the SCM model, to be used in model refinement
- `seSCMpar`: Standard errors of the parameter estimates in SCMpar

Details

Perform conditional maximum likelihood estimation of a VARMA model specified by the scalar component model approach, including the transformation matrix.

Value

- `data`: Observed time series
- `SCMs`: The specified SCMs
- `Tdx`: Indicator vector for the transformation matrix. The length of Tdx is k.
- `locTmtx`: Specification of estimable parameters of the transformation matrix
- `locAR`: Locators for the estimable parameters of the VAR coefficients
- `locMA`: Locators for the estimable parameters of the VMA coefficients
- `cnst`: A logical switch to include the constant vector in the model
coef  The parameter estimates
secoef  Standard errors of the parameter estimates
residuals  Residual series
Sigma  Residual covariance matrix
aic,bic  Information criteria of the fitted model
Phθ  Estimates of the constant vector, if any
Phi  Estimates of the VAR coefficients
Theta  Estimates of the VMA coefficients

Author(s)
Ruey S. Tsay

References

SCMid  Scalar Component Identification

Description
Find the overall order of a VARMA process via the scalar component model approach

Usage
SCMid(zt, maxp = 5, maxq = 5, h = 0, crit = 0.05, output = FALSE)

Arguments
zt  The T-by-k data matrix of a k-dimensional time series
maxp  Maximum AR order entertained. Default is 5.
maxq  Maximum MA order entertained. Default is 5.
h  The additional past lags used in canonical correlation analysis. Default is 0.
crit  Type-I error of the chi-square tests used.
output  A logical switch to control the output.

Value
Nmtx  The table of the numbers of zero canonical correlations
DDmtx  The diagonal difference table of the number of zero canonical correlations
Author(s)
Ruey S. Tsay

References

Examples
phi=matrix(c(0.2,-0.6,0.3,1.1),2,2); sigma=diag(2)
m1=VARMAsim(300,arlags=c(1),phi=phi,sigma=sigma)
zt=m1$series
m2=SCMid(zt)

SCMid2		Scalar Component Model Specification II

Description
Provides detailed analysis of scalar component models for a specified VARMA model. The overall model is specified by SCMid.

Usage
SCMid2(zt, maxp = 2, maxq = 2, h = 0, crit = 0.05, sseq = NULL)

Arguments
zt	The T-by-k data matrix of a k-dimensional time series
maxp	Maximum AR order specified. Default is 2.
maxq	Maximum MA order specified. Default is 2.
h	The additional past lags used in canonical correlation analysis. Default is zero.
crit	Type-I error used in testing. Default is 0.05.
sseq	The search sequence for SCM components. Default sequence starts with AR order.

Value
Tmatrix	The transformation matrix T
SCMorder	The orders of SCM components

Author(s)
Ruey S. Tsay
References


See Also

SCMid

Examples

phi=matrix(c(0.2,-0.6,0.3,1.1),2,2); sigma=diag(2)
m1=VARMA{sim}(300,arlags=c(1),phi=phi,sigma=sigma)
zt=m1$series
m2=SCMid2(zt)
names(m2)

SCMmod Scalar Component Model specification

Description

For a given set of SCMs and locator of transformation matrix, the program specifies a VARMA model via SCM approach for estimation.

Usage

SCMmod(order, ivor, output)

Arguments

order A k-by-2 matrix of the orders of SCM
ivor A k-dimensional vector indicating the location of "1" for each component in the transformation matrix.
output A logical switch to control output.

Details

The command specified estimable parameters for a VARMA model via the SCM components. In the output, "2" denotes estimation, "1" denotes fixing the value to 1, and "0" means fixing the parameter to zero.

Value

Tmtx Specification of the transformation matrix T
ARpar Specification of the VAR parameters
MApar Specification of the VMA parameters
**sVARMA**

**Author(s)**

Ruey S. Tsay

**References**


**Examples**

```r
ord=matrix(c(0,1,0,0),3,2)
Ivor=c(3,1,2)
ml=SCMmod(ord,Ivor,TRUE)
```

---

**sVARMA Seasonal VARMA Model Estimation**

**Description**

Performs conditional maximum likelihood estimation of a seasonal VARMA model

**Usage**

```r
sVARMA(da, order, sorder, s, include.mean = T, fixed = NULL, details = F, switch = F)
```

**Arguments**

- `da`: A T-by-k data matrix of a k-dimensional seasonal time series
- `order`: Regular order (p,d,q) of the model
- `sorder`: Seasonal order (P,D,Q) of the model
- `s`: Seasonality. s=4 for quarterly data and s=12 for monthly series
- `include.mean`: A logical switch to include the mean vector. Default is to include the mean
- `fixed`: A logical matrix to set zero parameter constraints
- `details`: A logical switch for output
- `switch`: A logical switch to exchange the ordering of the regular and seasonal VMA factors. Default is theta(B)^*Theta(B).

**Details**

Estimation of a seasonal VARMA model
Value

data  The data matrix of the observed k-dimensional time series
order  The regular order (p,d,q)
sorder  The seasonal order (P,D,Q)
period  Seasonality
cnst  A logical switch for the constant term
ccoef  Parameter estimates for use in model simplification
secoef  Standard errors of the parameter estimates
residuals  Residual series
Sigma  Residual covariance matrix
aic,bic  Information criteria of the fitted model
regPhi  Regular AR coefficients, if any
seaPhi  Seasonal AR coefficients
regTheta  Regular MA coefficients
seaTheta  Seasonal MA coefficients
Ph0  The constant vector, if any
switch  The logical switch to change the ordering of matrix product

Author(s)

Ruey S. Tsay

References


sVARMACpp  Seasonal VARMA Model Estimation (Cpp)

Description

Performs conditional maximum likelihood estimation of a seasonal VARMA model. This is the same function as sVARMA, with the likelihood function implemented in C++ for efficiency.

Usage

sVARMACpp(da, order, sorder, s, include.mean = T, fixed = NULL, details = F, switch = F)
Arguments

- **da**: A T-by-k data matrix of a k-dimensional seasonal time series
- **order**: Regular order (p,d,q) of the model
- **sorder**: Seasonal order (P,D,Q) of the model
- **s**: Seasonality. s=4 for quarterly data and s=12 for monthly series
- **include.mean**: A logical switch to include the mean vector. Default is to include the mean
- **fixed**: A logical matrix to set zero parameter constraints
- **details**: A logical switch for output
- **switch**: A logical switch to exchange the ordering of the regular and seasonal VMA factors. Default is theta(B)*Theta(B).

Details

Estimation of a seasonal VARMA model

Value

- **data**: The data matrix of the observed k-dimensional time series
- **order**: The regular order (p,d,q)
- **sorder**: The seasonal order (P,D,Q)
- **period**: Seasonality
- **cnst**: A logical switch for the constant term
- **ceof**: Parameter estimates for use in model simplification
- **secoef**: Standard errors of the parameter estimates
- **residuals**: Residual series
- **Sigma**: Residual covariance matrix
- **aic,bic**: Information criteria of the fitted model
- **regPhi**: Regular AR coefficients, if any
- **seaPhi**: Seasonal AR coefficients
- **regTheta**: Regular MA coefficients
- **seaTheta**: Seasonal MA coefficients
- **PhΘ**: The constant vector, if any
- **switch**: The logical switch to change the ordering of matrix product

Author(s)

Ruey S. Tsay

References

**sVARMApred**

*Prediction of a fitted multiplicative seasonal VARMA model*

**Description**

Perform prediction of a seasonal VARMA model

**Usage**

`sVARMApred(model, orig, h=1)`

**Arguments**

- **model**: An output of the `sVARMA` command
- **orig**: The forecast origin.
- **h**: The forecast horizon. For a given `h`, it computes 1-step to `h`-step ahead forecasts. Default is 1.

**Details**

Perform prediction of a fitted sVARMA model

**Value**

- **data**: The original data matrix
- **pred**: Forecasts
- **se.err**: Standard errors of forecasts
- **orig**: Return the forecast origin

**Author(s)**

Ruey S. Tsay

**References**

Description

Uses the diffusion index approach of Stock and Watson to compute out-of-sample forecasts

Usage

SWfore(y, x, orig, m)

Arguments

y The scalar variable of interest
x The data matrix (T-by-k) of the observed explanatory variables
orig Forecast origin
m The number of diffusion index used

Details

Performs PCA on X at the forecast origin. Then, fit a linear regression model to obtain the coefficients of prediction equation. Use the prediction equation to produce forecasts and compute forecast errors, if any. No recursive estimation is used.

Value

coef Regression coefficients of the prediction equation
yhat Predictions at the forecast origin
MSE Mean squared errors, if available
loadings Loading matrix
Dfindex Diffusion indices

Author(s)

Ruey S. Tsay

References

Description

Monthly simple returns of ten U.S. stocks. The sample period is from January 2001 to December 2011. Tick symbols of the ten stocks are used as column names for the returns.

Usage

tenstocks

Format

A 2-d list containing 132x11 observations.

Source

The original data were from Center for Research in Security Prices (CRSP) of the University of Chicago. The first column denotes the dates.

---

Transfer Function Model

Description

Estimates a transform function model. This program does not allow rational transfer function model. It is a special case of tfm1 and tfm2.

Usage

tfm(y, x, b = 0, s = 1, p = 0, q = 0)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>Data vector of dependent (output) variable</td>
</tr>
<tr>
<td>x</td>
<td>Data vector of independent variable</td>
</tr>
<tr>
<td>b</td>
<td>deadtime or delay</td>
</tr>
<tr>
<td>s</td>
<td>The order of the transfer function polynomial</td>
</tr>
<tr>
<td>p</td>
<td>AR order of the disturbance</td>
</tr>
<tr>
<td>q</td>
<td>MA order of the disturbance</td>
</tr>
</tbody>
</table>
Details

The model entertained is $y_t = c_0 + v(B)x_t + n_t$. $v(B) = 1 - v_1B - \ldots - v_sB^s$, and $n_t$ is an ARMA($p,q$) process.

Value

- $\text{coef}$: Coefficient estimates of the transfer function
- $\text{se.coef}$: Standard errors of the transfer function coefficients
- $\text{coef arma}$: Coefficient estimates of ARMA models
- $\text{se arma}$: Standard errors of ARMA coefficients
- $nt$: The disturbance series
- $\text{residuals}$: The residual series

Author(s)

Ruey S. Tsay

References


Description

Estimation of a general transfer function model. The model can only handle one input and one output.

Usage

tfm1(y, x, orderN, orderX)

Arguments

- $y$: Data vector of dependent variable
- $x$: Data vector of input (or independent) variable
- $\text{orderN}$: Order ($p,d,q$) of the disturbance component
- $\text{orderX}$: Order ($r,s,b$) of the transfer function model, where $r$ and $s$ are the degrees of denominator and numerator polynomials and $b$ is the delay

Details

Perform estimation of a general transfer function model
Value

estimate  Coefficient estimates
sigma2    Residual variance sigma-square
residuals Residual series
varcoef   Variance of the estimates
Nt        The disturbance series

Author(s)

Ruey S. Tsay

References


See Also

tfm

Examples

```r
# da=read.table("gasfur.txt")
# y=da[,2]; x=da[,1]
# m1=tfm(y,x,orderX=c(1,2,3),orderN=c(2,0,0))
```

---

**tfm2**  
*Transfer Function Model with Two Input Variables*

Description

Estimation of a general transfer function model with two input variables. The model can handle one output and up to 2 input variables. The time series noise can assume multiplicative seasonal ARMA models.

Usage

```r
tfm2(y,x,x2=NULL,ct=NULL,wt=NULL,orderN=c(1,0,0),orderS=c(0,0,0),
     sea=12,order1=c(0,1,0),order2=c(0,-1,0))
```
Arguments

- **y**: Data vector of dependent variable
- **x**: Data vector of the first input (or independent) variable
- **x2**: Data vector of the second input variable if any
- **ct**: Data vector of a given deterministic variable such as time trend, if any
- **wt**: Data vector of co-integrated series between input and output variables if any
- **orderN**: Order (p,d,q) of the regular ARMA part of the disturbance component
- **orderS**: Order (P,D,Q) of the seasonal ARMA part of the disturbance component
- **sea**: Seasonality, default is 12 for monthly data
- **order1**: Order (r,s,b) of the transfer function model of the first input variable, where r and s are the degrees of denominator and numerator polynomials and b is the delay
- **order2**: Order (r2,s2,b2) of the transfer function model of the second input variable, where 2r and s2 are the degrees of denominator and numerator polynomials and b2 is the delay

Details

Perform estimation of a general transfer function model with two input variables

Value

- **estimate**: Coefficient estimates
- **sigma2**: Residual variance sigma-square
- **residuals**: Residual series
- **varcoef**: Variance of the estimates
- **Nt**: The disturbance series
- **rAR**: Regular AR coefficients
- **rMA**: Regular MA coefficients
- **sAR**: Seasonal AR coefficients
- **sMA**: Seasonal MA coefficients
- **omega**: Numerator coefficients of the first transfer function
- **delta**: Denominator coefficients of the first transfer function
- **omega2**: Numerator coefficients of the 2nd transfer function
- **delta2**: Denominator coefficients of the 2nd transfer function

Author(s)

Ruey S. Tsay
References


See Also
tfm, tfm1

---

**VAR**

*Vector Autoregressive Model*

**Description**

Perform least squares estimation of a VAR model

**Usage**

\[
\text{VAR}(x, \ p = 1, \ output = T, \ include\_mean = T, \ fixed = \text{NULL})
\]

**Arguments**

- **x**: A T-by-k matrix of k-dimensional time series
- **p**: Order of VAR model. Default is 1.
- **output**: A logical switch to control output. Default is with output.
- **include\_mean**: A logical switch. It is true if mean vector is estimated.
- **fixed**: A logical matrix used in constrained estimation. It is used mainly in model simplification, e.g., removing insignificant estimates.

**Details**

To remove insignificant estimates, one specifies a threshold for individual t-ratio. The fixed matrix is then defined automatically to identify those parameters for removal.

**Value**

- **data**: Observed data
- **cnst**: A logical switch to include the mean constant vector
- **order**: VAR order
- **coef**: Coefficient matrix
- **aic, bic, hq**: Information criteria of the fitted model
- **residuals**: Residuals
- **secoef**: Standard errors of the coefficients to be used in model refinement
- **Sigma**: Residual covariance matrix
- **Phi**: AR coefficient polynomial
- **Ph\theta**: The constant vector
VARMA

Author(s)
Ruey S. Tsay

References

See Also
refVAR command

Examples

data("mts-examples", package="MTS")
gdp=log(gdp[,3:5])
zt=diff(gdp)
ml=VAR(zt,p=2)

Description
Performs conditional maximum likelihood estimation of a VARMA model. Multivariate Gaussian likelihood function is used.

Usage
VARMA(da, p = 0, q = 0, include.mean = T,
fixed = NULL, beta=NULL, sebeta=NULL,
prelim = F, details = F, thres = 2)

Arguments
da
    Data matrix (T-by-k) of a k-dimensional time series with sample size T.
p
    AR order
q
    MA order
include.mean
    A logical switch to control estimation of the mean vector. Default is to include
    the mean in estimation.
fixed
    A logical matrix to control zero coefficients in estimation. It is mainly used by
    the command refVARMA.
beta
    Parameter estimates to be used in model simplification, if needed
sebeta
    Standard errors of parameter estimates for use in model simplification
prelim
    A logical switch to control preliminary estimation. Default is none.
details
    A logical switch to control the amount of output.
thres
    A threshold used to set zero parameter constraints based on individual t-ratio. Default is 2.
Details

The fixed command is used for model refinement

Value

data Observed data matrix
ARorder VAR order
MAorder VMA order
cnst A logical switch to include the mean vector
coef Parameter estimates
secoef Standard errors of the estimates
residuals Residual matrix
Sigma Residual covariance matrix
aic,bic Information criteria of the fitted model
Phi VAR coefficients
Theta VMA coefficients
Ph0 The constant vector

Author(s)

Ruey S. Tsay

References


See Also

refVARMA

Examples

phi=matrix(c(0.2,-0.6,0.3,1.1),2,2); theta=matrix(c(-0.5,0,0,-0.5),2,2)
sigma=diag(2)
m1=VARMA(sum(300,arlags=c(1),malags=c(1),phi=phi,theta=theta,sigma=sigma)
zV=m1$series
m2=VARMA(zV,p=1,q=1,include.mean=FALSE)
VARMAcov

Parking Matrices of a VARMA Model

Description

Uses psi-weights to compute the autocovariance matrices of a VARMA model.

Usage

VARMAcov(Phi = NULL, Theta = NULL, Sigma = NULL, lag = 12, trun = 120)

Arguments

- **Phi**: A k-by-kp matrix consisting of VAR coefficient matrices, \( \Phi = [\Phi_1, \Phi_2, ..., \Phi_p] \).
- **Theta**: A k-by-kq matrix consisting of VMA coefficient matrices, \( \Theta = [\Theta_1, \Theta_2, ..., \Theta_q] \).
- **Sigma**: Covariance matrix of the innovations (k-by-k).
- **lag**: Number of cross-covariance matrices to be computed. Default is 12.
- **trun**: The lags of psi-weights used in calculation. Default is 120.

Details

Use psi-weight matrices to compute approximate autocovariance matrices of a VARMA model.

Value

- **autocov**: Autocovariance matrices
- **ccm**: Auto correlation matrices

Author(s)

Ruey S. Tsay

References


Examples

```r
Phi=matrix(c(0.2,-0.6,0.3,1.1),2,2)
Sig=matrix(c(4,1,1,1),2,2)
VARMAcov(Phi=Phi,Sigma=Sig)
```
VARMACpp  

Vector Autoregressive Moving-Average Models (Cpp)

Description

Performs conditional maximum likelihood estimation of a VARMA model. Multivariate Gaussian likelihood function is used. This is the same function as VARMA, with the likelihood function implemented in C++ for efficiency.

Usage

VARMACpp(da, p = 0, q = 0, include.mean = T,
       fixed = NULL, beta=NULL, sebeta=NULL,
       prelim = F, details = F, thres = 2)

Arguments

da  Data matrix (T-by-k) of a k-dimensional time series with sample size T.
p  AR order
q  MA order
include.mean  A logical switch to control estimation of the mean vector. Default is to include the mean in estimation.
fixed  A logical matrix to control zero coefficients in estimation. It is mainly used by the command refVARMA.
beta  Parameter estimates to be used in model simplification, if needed
sebeta  Standard errors of parameter estimates for use in model simplification
prelim  A logical switch to control preliminary estimation. Default is none.
details  A logical switch to control the amount of output.
thres  A threshold used to set zero parameter constraints based on individual t-ratio. Default is 2.

Details

The fixed command is used for model refinement

Value

data  Observed data matrix
ARorder  VAR order
MAorder  VMA order
cnst  A logical switch to include the mean vector
coef  Parameter estimates
secoef  Standard errors of the estimates
Residuals: Residual matrix
Sigma: Residual covariance matrix
aic, bic: Information criteria of the fitted model
Phi: VAR coefficients
Theta: VMA coefficients
Phθ: The constant vector

Author(s)
Ruey S. Tsay

References

See Also
VARMA

Examples
phi=matrix(c(0.2,-0.6,0.3,1.1),2,2); theta=matrix(c(-0.5,0,0,-0.5),2,2)
sigma=diag(2)
m1=VARMasim(300,arlags=c(1),malags=c(1),phi=phi,theta=theta,sigma=sigma)
zt=m1$series
m2=VARMA(zt,p=1,q=1,include.mean=FALSE)

Description
Compute and plot the impulse response function of a given VARMA model

Usage
VARMAirf(Phi = NULL, Theta = NULL, Sigma = NULL, lag = 12, orth = TRUE)

Arguments
Phi: A k-by-kp matrix of VAR coefficients in the form Phi=[Φ1, Φ2, ..., Φp].
Theta: A k-by-kq matrix of VMA coefficients in the form Θ=[Θ1, Θ2, ..., Θq]
Sigma: Covariance matrix (k-by-k) of the innovations.
lag: Number of lags of impulse response functions to be computed
orth: A logical switch to use orthogonal innovations. Default is to perform orthogonalization of the innovations.
VARMApred

Value

psi          The Psi-weight matrices
irf         Impulse response functions

Author(s)

Ruey S. Tsay

References


See Also

VARMApsi command

Examples

p1=matrix(c(0.2,-0.6,0.3,1.1),2,2)
th1=matrix(c(-0.5,0.2,0.0,-0.6),2,2)
Sig=matrix(c(4,1,1,2,2)
ml=VARMAirf(Phi=p1,Theta=th1,Sigma=Sig)

VARMApred

VARMA Prediction

Description

Compute forecasts and their associate forecast error covariances of a VARMA model

Usage

VARMApred(model, h = 1, orig = 0)

Arguments

model          A fitted VARMA model
h           Number of steps of forecasts, i.e., forecast horizon.
orig          Forecast origin. Default is the end of the sample.

Value

pred          Predictions
se.err        Standard errors of forecasts
orig           Forecast origin
VARMAsim

Author(s)
Ruey S. Tsay

References

VARMAsim

Generating a VARMA Process

Description
Performs simulation of a given VARMA model

Usage
VARMAsim(nobs, arlags = NULL, malags = NULL, cnst = NULL, phi = NULL, theta = NULL, skip = 200, sigma)

Arguments
nobs Sample size
arlags The exact lags of the VAR matrix polynomial.
malags The exact lags of the VMA matrix polynomial.
cnst Constant vector, Phi0
phi Matrix of VAR coefficient matrices in the order of the given arlags.
theta Matrix of VMA coefficient matrices in the order of the given malags.
skip The number of initial data to be omitted. Default is 200.
sigma Covariance matrix (k-by-k, positive definite) of the innovations

Details
Use multivariate Gaussian distribution to generate random shocks. Then, generate a given VARMA model. The first skip data points were discarded.

Value
series Generated series
noises The noise series

Author(s)
Ruey S. Tsay
References


Examples

```r
p1 = matrix(c(0.2, -0.6, 0.3, 1, 1), 2, 2)
sig = matrix(c(4, 0.8, 0.8, 1), 2, 2)
th1 = matrix(c(-0.5, 0, 0, -0.6), 2, 2)
m1 = VARMA.sim(300, arlags = c(1), malags = c(1), phi = p1, theta = th1, sigma = sig)
zt = m1$series
```

<table>
<thead>
<tr>
<th>VAROrder</th>
<th>VAR Order Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

Computes information criteria and the sequential Chi-square statistics for a vector autoregressive process

Usage

VAROrder(x, maxp = 13, output = T)

Arguments

- `x`: Data matrix of dimension T-by-k with T being the sample size and k the number of time series
- `maxp`: The maximum VAR order entertained. Default is 13.
- `output`: A logical switch to control the output. Default is to provide output

Details

For a given maxp, the command computes Akaike, Bayesian and Hannan-Quinn information criteria for various VAR models using the data from t=maxp+1 to T. It also computes the Tiao-Box sequential Chi-square statistics and their p-values.

Value

- `aic`: Akaike information criterion
- `bic`: Bayesian information criterion
- `hq`: Hannan and Quinn information criterion
- `aicor, bicor, hqor`: Orders selected by various criteria
- `Mstat`: Chi-square test statistics
- `Mpv`: p-values of the Mstat
VARorderI

Author(s)

Ruey S. Tsay

References


See Also

VARorderI

Examples

data("mts-examples",package="MTS")
zt=diffM(log(qgdp[,]3:5))
VARorderI(zt,maxp=8)

VARorderI VAR order specification I

Description

This program is similar to VARorder, but it uses observations from t=p+1 to T to compute the information criteria for a given VAR(p) model.

Usage

VARorderI(x, maxp = 13, output = T)

Arguments

x A T-by-k data matrix of vector time series
maxp The maximum VAR order entertained
output A logical switch to control output

Details

For a given VAR(p) model, the program uses observations from t=p+1 to T to compute the information criteria. Therefore, different numbers of data points are used to estimate different VAR models.
Value

- `aic`: Akaike information criterion
- `aicor`: Order selected by AIC
- `bic`: Bayesian information criterion
- `bicor`: Order selected by BIC
- `hq`: Hannan and Quinn information criterion
- `hqor`: Order selected by hq
- `Mstat`: Step-wise Chi-square statistics
- `mpv`: p-values of the M-statistics

Author(s)

Ruey S Tsay

References

Tsay (2014)

See Also

VARorder

Description

Computes the forecasts of a VAR model, the associated standard errors of forecasts and the mean squared errors of forecasts

Usage

```r
VARpred(model, h = 1, orig = 0, Out.level = FALSE, output = TRUE)
```

Arguments

- `model`: An output object of a VAR or re VAR command
- `h`: Forecast horizon, a positive integer
- `orig`: Forecast origin. Default is zero meaning the forecast origin is the last data point
- `Out.level`: Boolean control for details of output
- `output`: Boolean control for printing forecast results

Details

Computes point forecasts and the associated variances of forecast errors
VARpsi

Value
pred   Point predictions
se.err Standard errors of the predictions
mse    Mean-square errors of the predictions

Author(s)
Ruey S. Tsay

References

Examples
data("mts-examples", package="MTS")
gdp=log(ogdp[,3:5])
zt=diff(gdp)
ml=VAR(zt, p=2)
VARpred(ml, 4)

---

VARpsi  

VAR Psi-weights

Description
Computes the psi-weight matrices of a VAR model

Usage
VARpsi(Phi, lag = 5)

Arguments
Phi      A k-by-kp matrix of VAR coefficients in the form Phi=[Phi1, Phi2, ..., Phip]
lag     Number of psi-weight lags

Value
Psi-weights of a VAR model

Author(s)
Ruey S. Tsay
References


Examples

```r
p1 = matrix(c(0.2, -0.6, 0.3, 1.1), 2, 2)
m1 = VARpsi(p1, 4)
names(m1)
```

---

**VARs**  
*VAR Model with Selected Lags*

**Description**

This is a modified version of VAR command by allowing the users to specify which AR lags to be included in the model.

**Usage**

```r
VARs(x, lags, include.mean = T, output = T, fixed = NULL)
```

**Arguments**

- **x**  
  A T-by-k data matrix of k-dimensional time series with T observations
- **lags**  
  A vector of non-zero AR lags. For instance, lags=c(1,3) denotes a VAR(3) model with Phi2 = 0.
- **include.mean**  
  A logical switch to include the mean vector
- **output**  
  A logical switch to control output
- **fixed**  
  A logical matrix to fix parameters to zero.

**Details**

Performs VAR estimation by allowing certain lag coefficient matrices being zero.

**Value**

- **data**  
  Observed time series data
- **lags**  
  The selected VAR lags
- **order**  
  The VAR order
- **cnst**  
  A logical switch to include the mean vector
- **coef**  
  Parameter estimates
- **aic,bic**  
  Information criteria of the fitted model
- **residuals**  
  Residual series
secoef: Standard errors of the estimates
Sigma: Residual covariance matrix
Phi: VAR coefficient matrix
Ph0: A constant vector

Author(s)
Ruey S. Tsay

References

See Also
VAR command

Examples
```
data("mts-examples", package="MTS")
zt=log(qgdp[,3:5])
m1=VARs(zt, lags=c(1,2,4))
```

VARX

**VAR Model with Exogenous Variables**

**Description**
Estimation of a VARX model

**Usage**
```
VARX(zt, p, xt = NULL, m = 0, include.mean = T, fixed = NULL, output = T)
```

**Arguments**
- `zt`: A T-by-k data matrix of a k-dimensional time series
- `p`: The VAR order
- `xt`: A T-by-kx data matrix of kx exogenous variables
- `m`: The number of lags of exogenous variables
- `include.mean`: A logical switch to include the constant vector. Default is to include the constant.
- `fixed`: A logical matrix for setting parameters to zero.
- `output`: A logical switch to control output
**Details**

Performs least squares estimation of a VARX(p,s) model

**Value**

- **data**: The observed time series
- **xt**: The data matrix of explanatory variables
- **aror**: VAR order
- **m**: The number of lags of explanatory variables used
- **Phi**: The constant vector
- **Phi**: VAR coefficient matrix
- **beta**: The regression coefficient matrix
- **residuals**: Residual series
- **coef**: The parameter estimates to be used in model simplification
- **se.coef**: Standard errors of the parameter estimates
- **include.mean**: A logical switch to include the mean vector

**Author(s)**

Ruey S. Tsay

**References**


---

**VARXirf**

*Impluse response function of a fitted VARX model*

**Description**

Compute the impulse response functions and cumulative impulse response functions of a fitted VARX model

**Usage**

`VARXirf(model, lag=12, orth=TRUE)`

**Arguments**

- **model**: An output of the VARX (or refVARX) command for a vector time series with exogeneous variables
- **lag**: The number of lags of the impulse response function to be computed. Default is 12.
- **orth**: The control variable for using orthogonal innovations. This command applies to the impulse response functions of the VAR part only.
VARXorder

Details

Compute the impulse response functions and cumulative impulse response functions of a fitted VARX model. The impulse response function of the exogeneous variables are also given. The plots of impulse response functions are shown.

Value

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>irf</td>
<td>Impulse response functions of the VAR part, original innovations used</td>
</tr>
<tr>
<td>orthirf</td>
<td>Impulse response functions of the VAR part using orthogonal innovations</td>
</tr>
<tr>
<td>irfx</td>
<td>Impulse response function of the exogenous variables</td>
</tr>
</tbody>
</table>

Author(s)

Ruey S. Tsay

References


VARXorder

| VARXorder | VARX Order Specification |

Description

Specifies the orders of a VARX model, including AR order and the number of lags of exogenous variables

Usage

VARXorder(x, exog, maxp = 13, maxm = 3, output = T)

Arguments

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>A T-by-k data matrix of a k-dimensional time series</td>
</tr>
<tr>
<td>exog</td>
<td>A T-by-v data matrix of exogenous variables</td>
</tr>
<tr>
<td>maxp</td>
<td>The maximum VAR order entertained</td>
</tr>
<tr>
<td>maxm</td>
<td>The maximum lags of exogenous variables entertained</td>
</tr>
<tr>
<td>output</td>
<td>A logical switch to control output</td>
</tr>
</tbody>
</table>

Details

Computes the information criteria of a VARX process
Value

- `aic`: Akaike information criterion
- `aicor`: Order selected by AIC
- `bic`: Bayesian information criterion
- `bicor`: Order selected by BIC
- `hq`: Hannan and Quinn information criterion
- `hqor`: Order selected by hq

Author(s)

Ruey S. Tsay

References


VARXpred (VARX Model Prediction)

Description

Computes point forecasts of a VARX model. The values of exogenous variables must be given.

Usage

```r
VARXpred(m1, newxt = NULL, hstep = 1, orig = 0)
```

Arguments

- `m1`: An output object of VARX or refVARX command
- `newxt`: The data matrix of exogenous variables needed in forecasts.
- `hstep`: Forecast horizon
- `orig`: Forecast origin. Default is 0, meaning the last data point.

Details

Uses the provided exogenous variables and the model to compute forecasts

Value

Point forecasts and their standard errors

Author(s)

Ruey S. Tsay
References


---

**Vech**

*Half-Stacking Vector of a Symmetric Matrix*

**Description**

Obtain the half-stacking vector of a symmetric matrix

**Usage**

```r
Vech(mtx)
```

**Arguments**

- `mtx` A symmetric matrix

**Details**

Stacking a matrix into a vector using data on and below the diagonal.

**Value**

a vector consisting of stacked elements of a symmetric matrix

**Author(s)**

Ruey S. Tsay

**Examples**

```r
m1=matrix(c(1:9),3,3)
m2=(m1+t(m1))/2
v1=Vech(m2)
```
Matrix constructed from output of the Vech Command. In other words, restore the original symmetric matrix from its half-stacking vector.

Description
Restores the symmetric matrix from the Vech command

Usage
VechM(vec)

Arguments
vec A vector representing the half-stacking of a symmetric matrix

Details
This command re-construct a symmetric matrix from output of the Vech command

Value
A symmetric matrix

Author(s)
Ruey S. Tsay

References
Tsay (2014, Appendix A)

See Also
Vech

Examples
v1=c(2,1,3)
m1=VechM(v1)
m1
VMA

Vector Moving Average Model

Description

Performs VMA estimation using the conditional multivariate Gaussian likelihood function

Usage

VMA(da, q = 1, include.mean = T, fixed = NULL,
    beta=NULL, sebeta=NULL, prelim = F,
    details = F, thres = 2)

Arguments

da             Data matrix of a k-dimensional VMA process with each column containing one time series
q               The order of VMA model
include.mean   A logical switch to include the mean vector. The default is to include the mean vector in estimation.
fixed          A logical matrix used to fix parameter to zero
beta           Parameter estimates for use in model simplification
sebeta         Standard errors of parameter estimates for use in model simplification
prelim         A logical switch to select parameters to be included in estimation
details        A logical switch to control the amount of output
thres          Threshold for t-ratio used to fix parameter to zero. Default is 2.

Value

data            The data of the observed time series
MAorder         The VMA order
cnst            A logical switch to include the mean vector
decoef          Parameter estimates
secoef          Standard errors of the parameter estimates
residuals       Residual series
Sigma           Residual covariance matrix
Theta           The VAR coefficient matrix
mu              The constant vector
aic,bic         The information criteria of the fitted model

Author(s)

Ruey S. Tsay
References

Tsay (2014, Chapter 3).

Examples

```r
theta = matrix(c(0.5, 0.4, 0.6), 2, 2); sigma = diag(2)
m1 = VARMA.sim(200, malags = c(1), theta = theta, sigma = sigma)
zt = m1$series
m2 = VMA(zt, q = 1, include.mean = FALSE)
```

---

### VMACpp

Vector Moving Average Model (Cpp)

#### Description

Performs VMA estimation using the conditional multivariate Gaussian likelihood function. This is the same function as VMA, with the likelihood function implemented in C++ for efficiency.

#### Usage

```r
VMACpp(da, q = 1, include.mean = T, fixed = NULL, beta = NULL, sebeta = NULL, prelim = F, details = F, thres = 2)
```

#### Arguments

- **da**: Data matrix of a k-dimensional VMA process with each column containing one time series
- **q**: The order of VMA model
- **include.mean**: A logical switch to include the mean vector. The default is to include the mean vector in estimation.
- **fixed**: A logical matrix used to fix parameter to zero
- **beta**: Parameter estimates for use in model simplification
- **sebeta**: Standard errors of parameter estimates for use in model simplification
- **prelim**: A logical switch to select parameters to be included in estimation
- **details**: A logical switch to control the amount of output
- **thres**: Threshold for t-ratio used to fix parameter to zero. Default is 2.
Value

data The data of the observed time series
MAorder The VMA order
cnst A logical switch to include the mean vector
coeff Parameter estimates
secoeft Standard errors of the parameter estimates
residuals Residual series
Sigma Residual covariance matrix
Theta The VAR coefficient matrix
mu The constant vector
aic bic The information criteria of the fitted model

Author(s)
Ruey S. Tsay

References
Tsay (2014, Chapter 3).

See Also
VMA

Examples
theta = matrix(c(0.5, 0.4, 0.6, 2, 2), sigma = diag(2)
m1 = VARMAsim(200, mlag = c(1), theta = theta, sigma = sigma)
zt = m1$series
m2 = VMACpp(zt, q = 1, include.mean = FALSE)

VMAe VMA Estimation with Exact likelihood

Description
Estimation of a VMA(q) model using the exact likelihood method. Multivariate Gaussian likelihood function is used.

Usage
VMAe(data, q = 1, include.mean = T, coef0 = NULL,
      secoeft = NULL, fixed = NULL, prelim = F,
      details = F, thres = 2)
Arguments

da         Data matrix (T-by-k) for a k-dimensional VMA process
q          The order of a VMA model
include.mean A logical switch to include the mean vector in estimation. Default is to include
             the mean vector.
coef0       Initial estimates of the coefficients used mainly in model refinement
secoef0     Standard errors of the initial estimates
fixed       A logical matrix to put zero parameter constraints
prelim      A logical switch for preliminary estimation
details     A logical switch to control output in estimation
thres       The threshold value for zero parameter constraints

Value

data        The observed time series
MAorder     The VMA order
cnst        A logical switch to include the mean vector
coef        Parameter estimates
secoef      Standard errors of parameter estimates
residuals   Residual series
Sigma       Residual covariance matrix
Theta       VMA coefficient matrix
mu          The mean vector
aic,bic     The information criteria of the fitted model

Author(s)

Ruey S. Tsay

References

Hoboken, NJ.

See Also

VMA
**VMAorder**

**VMA Order Specification**

**Description**
Performs multivariate Ljung-Box tests to specify the order of a VMA process

**Usage**

VMAorder(x, lag = 20)

**Arguments**

- **x**: Data matrix of the observed k-dimensional time series. Each column represents a time series.
- **lag**: The maximum VMA order entertained. Default is 20.

**Details**
For a given lag, the command computes the Ljung-Box statistic for testing \( \rho_j = \ldots = \rho_{\text{lag}} = 0 \), where \( j = 1, 2, \ldots, \text{lag} \). For a VMA(q) process, the Ljung-Box statistics should be significant for the first \( q \) lags, and insignificant thereafter.

**Value**
The Q-statistics and p-value plot

**Author(s)**
Ruey S. Tsay

**References**

**Examples**

zt = matrix(rnorm(600), 200, 3)
VMAorder(zt)
Description

Performs the conditional maximum likelihood estimation of a VMA model with selected lags in the model.

Usage

VMAs(da, malags, include.mean = T, fixed = NULL, prelim = F, details = F, thres = 2)

Arguments

da - A T-by-k matrix of a k-dimensional time series with T observations
malags - A vector consisting of non-zero MA lags
include.mean - A logical switch to include the mean vector
fixed - A logical matrix to fix coefficients to zero
prelim - A logical switch concerning initial estimation
details - A logical switch to control output level
thres - A threshold value for setting coefficient estimates to zero

Details

A modified version of VMA model by allowing the user to select non-zero MA lags.

Value

data - The observed time series
Malags - The VMA lags
cnst - A logical switch to include the mean vector
coef - The parameter estimates
secoef - The standard errors of the estimates
residuals - Residual series
aic, bic - The information criteria of the fitted model
Sigma - Residual covariance matrix
Theta - The VMA matrix polynomial
mu - The mean vector
MOrder - The VMA order

Author(s)

Ruey S. Tsay
Vmiss

References

See Also
  VMA

Vmiss

VARMA Model with Missing Value

Description
  Assuming that the model is known, this program estimates the value of a missing data point. The whole data point is missing.

Usage
  Vmiss(zt, piwgt, sigma, tmiss, cnst = NULL, output = T)

Arguments
  zt                  A T-by-k data matrix of a k-dimensional time series
  piwgt              The pi-weights of a VARMA model defined as piwgt=[pi0, pi1, pi2, ....]
  sigma              Positive definite covariance matrix of the innovations
  tmiss              Time index of the missing data point
  cnst               Constant term of the model
  output             A logical switch to control output

Details
  Use the least squares method to estimate a missing data point. The missing is random.

Value
  Estimates of the missing values

Author(s)
  Ruey S. Tsay

References
See Also

Vpmiss

Examples

data("mts-examples", package="MTS")
gdp=log(ogdp[,3:5])
m1=VAR(gdp,3)
piwgt=m1$Phi; Sig=m1$Sigma; cnst=m1$Ph0
m2=Vmiss(gdp,piwgt,Sig,50,cnst)

Vpmiss  Partial Missing Value of a VARMA Series

Description

Assuming that the data is only partially missing, this program estimates those missing values. The model is assumed to be known.

Usage

Vpmiss(zt, piwgt, sigma, tmiss, mdx, cnst = NULL, output = T)

Arguments

zt  A T-by-k data matrix of a k-dimensional time series
piwgt  pi-weights of the model in the form piwgt[pi0, pi1, pi2, ....]
sigma  Residual covariance matrix
tmiss  Time index of the partially missing data point
mdx  A k-dimensional indicator with "0" denoting missing component and "1" denoting observed value.
cnst  Constant term of the model
output  values of the partially missing data

Value

Estimates of the missing values

Author(s)

Ruey S. Tsay

References

See Also

Vmiss

Examples

#data("mts-examples", package = "MTS")
#gdp = log(gdp[, 3:5])
#m1 = VAR(gdp, 1)
#piwgt = m1$Phi; cnst = m1$Ph0; Sig = m1$Sigma
#mdx = c(0, 1, 1)
#m2 = Vpmiss(gdp, piwgt, Sig, 50, mdx, cnst)
Index

*Topic datasets
  ibmspko, 24
  qgdp, 41
  tenstocks, 64

  apca, 4
  archTest, 5

  backtest, 6
  BEKK11, 7
  Btfm2, 8
  BVAR, 9

  ccm, 11
  comVol, 12
  Corner, 13

  dccFit, 13
  dccPre, 14
  diffM, 16

  Eccm, 16
  ECMvar, 17
  ECMvar1, 19
  EWMAvol, 20

  FEVdec, 21

  GrangerTest, 22

  hfactor, 23
  ibmspko, 24

  Kronfit, 24
  Kronid, 26
  Kronpred, 27
  Kronspec, 28

  Lminv (MTS-internal), 36

  MarchTest, 29

  MCHdiag, 31
  MCholY, 32
  mFilter (MTS-internal), 36
  Mlm, 33
  mq, 33
  msqrt, 34
  mtCopula, 35
  MTS (MTS-package), 4
  MTS-internal, 36
  MTS-package, 4
  MTSdiag, 36
  MTStplot, 37
  MtXprod, 38
  MtXprod1, 38

  PIwgt, 39
  PSIwgt, 40

  qgdp, 41

  refECMVvar, 41
  refECMVvar1, 42
  refKronfit, 43
  refREGts, 44
  refSCMfit, 44
  refsVARMA, 45
  refVAR, 46
  refVARMA, 47
  refVARs (MTS-internal), 36
  refVARX, 48
  refVMA, 49
  refVMAe, 50
  refVMAs (MTS-internal), 36
  REGts, 51
  REGtspred, 52
  revmq (MTS-internal), 36
  RLS, 53

  SCCor, 54
  SCMfit, 55
INDEX

SCMid, 56
SCMid2, 57
SCMmod, 58
sVARMA, 59
sVARMACpp, 60
sVARMAPred, 62
SWfore, 63

tenstocks, 64
tfm, 64
tfm1, 65
tfm2, 66

VAR, 68
VARChii (MTS-internal), 36
VARRecm (MTS-internal), 36
VARfore (MTS-internal), 36
VARirf (MTS-internal), 36
VARMA, 69
VARMAcov, 71
VARMACpp, 72
VARMAirf, 73
VARMAPred, 74
VARMAsim, 75
VAROrder, 76
VAROrderI, 77
VARpred, 78
VARpsi, 79
VARs, 80
VARX, 81
VARXirf, 82
VARXorder, 83
VARXpred, 84
Vech, 85
VechM, 86
VMA, 87
VMACpp, 88
VMAe, 89
VMAorder, 91
VARApred (MTS-internal), 36
VMAs, 92
VMiss, 93
VPmiss, 94