Package ‘portes’

June 19, 2023

Type Package

Title Portmanteau Tests for Time Series Models

Version 6.0

Author Esam Mahdi

Maintainer Esam Mahdi <emahdi2012@gmail.com>

Description Contains common univariate and multivariate portmanteau test statistics for time series models. These tests are based on using asymptotic distributions such as chi-square distribution and based on using the Monte Carlo significance tests. Also, it can be used to simulate from univariate and multivariate seasonal time series models.

Imports parallel, forecast(>= 8.21), vars(>= 1.5-9)

License GPL (>= 2)

Encoding UTF-8

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2023-06-18 22:20:02 UTC

R topics documented:

`portes-package` *Portmanteau Tests for Time Series Models*

Description

This package contains a set of portmanteau diagnostic checks for univariate and multivariate time series based on the asymptotic approximation distributions and the Monte-Carlo significance test. More details about the portmanteau test statistics are available online from the vignette of this package. This package can be also used to simulate univariate and multivariate data from seasonal/nonseasonal `ARIMA/ VARIMA` models with innovations from finite or infinite variances from stable distributions. The simulated data may have deterministic terms, constant drifts and time trends, with non-zero means.

Details

Package:	portes
Type:	Package
Version:	6.0
Date:	2023-06-06
LazyLoad:	yes
LazyData:	yes
Classification/ACM:	G.3, G.4, I.5.1
Classification/MSC:	62M10, 91B84
License:	GPL (>= 2)

Author(s)

Esam Mahdi and A. Ian McLeod.

Maintainer: Esam Mahdi <emahdi2012@gmail.com>

BoxPierce *The Univariate-Multivariate Box and Pierce Portmanteau Test*

Description

The univariate or multivariate Box-Pierce (1970) portmanteau test.

Usage

```
BoxPierce(obj,lags=seq(5,30,5),fitdf=0,sqrd.res=FALSE)
```

Arguments

`obj`	a univariate or multivariate series with class `"numeric"`, `"matrix"`, `"ts"`, or (`"mts"` `"ts"`). It can be also an object of fitted time-series model with class `"ar"`, `"arima0"`, `"Arima"`, (`"ARIMA forecast ARIMA Arima"`), `"lm"`, (`"glm"` `"lm"`), or `"varest"`. `obj` may also an object with class `"list"` (see details and following examples).
`lags`	vector of lag auto-cross correlation coefficients used for `Hosking` test.
`fitdf`	Default is zero for testing the randomness of a given sequence with class `"numeric"`, `"matrix"`, `"ts"`, or (`"mts"` `"ts"`). In general `fitdf` equals to the number of estimated parameters in the fitted model. If `obj` is an object with class `"ar"`, `"arima0"`, `"Arima"`, `"varest"`, (`"ARIMA forecast ARIMA Arima"`), or `"list"` then no need to enter the value of `fitdf` as it will be automatically determined. For `obj` with other classes, the `fitdf` is needed for degrees of freedom of asymptotic chi-square distribution.
`sqrd.res`	if `TRUE` then apply the test on the squared values. This checks for Autoregressive Conditional Heteroscedastic, `ARCH`, effects. When `sqrd.res = FALSE`, then apply the test on the usual residuals.

Details

However the portmanteau test statistic can be applied directly on the output objects from the built in `R` functions `ar()`, `ar.ols()`, `ar.burg()`, `ar.yw()`, `ar.mle()`, `arima()`, `arim0()`, `Arima()`, `auto.arima()`, `lm()`, `glm()`, and `VAR()`, it works with output objects from any fitted model. In this case, users should write their own function to fit any model they want, where they may use the built in `R` functions `garch()`, `garchFit()`, `fracdiff()`, `tar()`, etc. The object `obj` represents the output of this function. This output must be a list with at least two outcomes: the fitted residual and the fitdf of the fitted model (`list(res = ..., fitdf = ...)`). See the following example with the function `FitModel()`.

Note: In `stats` `R`, the function `Box.test` was built to compute the Box and Pierce (1970) and Ljung and Box (1978) test statistics only in the univariate case where we can not use more than one single lag value at a time. The functions `BoxPierce` and `LjungBox` are more accurate than `Box.test` function and can be used in the univariate or multivariate time series at vector of different lag values as well as they can be applied on an output object from a fitted model described in the description of the function `BoxPierce`.

Value

The Box and Pierce univariate or multivariate test statistic with the associated p-values for different lags based on the asymptotic chi-square distribution with `k^2(lags-fitdf)` degrees of freedom.

Author(s)

Esam Mahdi and A.I. McLeod.

References

Box, G.E.P. and Pierce, D.A. (1970). "Distribution of Residual Autocorrelation in Autoregressive-Integrated Moving Average Time Series Models". Journal of American Statistical Association, 65, 1509-1526.

See Also

`Box.test`, `LjungBox`, `MahdiMcLeod`, `Hosking`, `LiMcLeod`, `portest`, `GetResiduals`.

Examples

```
x <- rnorm(100)
BoxPierce(x)                              ## univariate test
x <- cbind(rnorm(100),rnorm(100))
BoxPierce(x)                              ## multivariate test
##
##
## Annual flow of the river Nile at Aswan - 1871 to 1970
fit <- arima(Nile, c(1, 0, 1))
lags <- c(5, 10, 20)
## Apply the univariate test statistic on the fitted model
BoxPierce(fit, lags)            ## Correct (no need to specify fitdf)
BoxPierce(fit, lags, fitdf = 2) ## Correct
## Apply the test statistic on the residuals and set fitdf = 2
res <- resid(fit)
BoxPierce(res, lags)             ## Wrong (fitdf is needed!)
BoxPierce(res, lags, fitdf = 2)  ## Correct
##
##
## Quarterly, west German investment, income, and consumption from 1960 Q1 to 1982 Q4
data(WestGerman)
DiffData <- matrix(numeric(3 * 91), ncol = 3)
  for (i in 1:3)
    DiffData[, i] <- diff(log(WestGerman[, i]), lag = 1)
fit <- ar.ols(DiffData, intercept = TRUE, order.max = 2)
lags <- c(5,10)
## Apply the test statistic on the fitted model
BoxPierce(fit,lags)                ## Correct (no need to specify fitdf)
## Apply the test statistic on the residuals where fitdf = 2
res <- ts(na.omit(fit$resid))
BoxPierce(res,lags)                ## Wrong (fitdf is needed!)
BoxPierce(res,lags,fitdf = 2)      ## Correct
```

```
##
##
## Monthly log stock returns of Intel corporation data: Test for ARCH Effects
monthintel <- as.ts(monthintel)
BoxPierce(monthintel)                          ## Usual test
BoxPierce(monthintel,sqrd.res=TRUE)  ## Test for ARCH effects
##
#### Write a function to fit a model: Apply portmanteau test on fitted obj with class "list"
## Example
FitModel <- function(data){
    fit <- ar.ols(data, intercept = TRUE, order.max = 2)
    fitdf <- 2
    res <- ts(na.omit(fit$resid))
 list(res=res,fitdf=fitdf)
}
data(WestGerman)
DiffData <- matrix(numeric(3 * 91), ncol = 3)
  for (i in 1:3)
    DiffData[, i] <- diff(log(WestGerman[, i]), lag = 1)
Fit <- FitModel(DiffData)
BoxPierce(Fit)
```

CRSP *Monthly simple returns of the CRSP value-weighted index, 1926 to 1997*

Description

This data has been discussed by Tsay (2002, Ch.2, p.38 and 39) and Lin and McLeod (2008). There are 864 data values.

Usage

```
data(CRSP)
```

References

Lin, J.-W. and McLeod, A.I. (2008). "Portmanteau Tests for ARMA Models with Infinite Variance". Journal of Time Series Analysis, 29, 600-617.

Tsay, R. S. (2002). Analysis of Financial Time Series. Wiley, New York.

Examples

```
acf(CRSP)
```

`DEXCAUS` *Canada/US Foreign Exchanges Rates, Daily, Jan. 4, 1971 to Sept. 5, 1996.*

Description

There are 2513 data values.

Usage

```
data(DEXCAUS)
```

Details

Title: Canada / U.S. Foreign Exchange Rate Series ID: DEXCAUS Source: Board of Governors of the Federal Reserve System Release: H.10 Foreign Exchange Rates Seasonal Adjustment: Not Applicable Frequency: Daily Units: Canadian Dollars to One U.S. Dollar Date Range: 1971-01-04 to 2006-09-05 Last Updated: 2006-09-06 10:42 AM CT Notes: Noon buying rates in New York City for cable transfers payable in foreign currencies.

Source

https://fred.stlouisfed.org/series/DEXCAUS

Examples

```
acf(DEXCAUS)
```

`EconomicUK` *Quarterly U.K. economic time series from 1957 Q3 to 1967 Q4*

Description

The data are quarterly, seasonally unadjusted in 1958 prices, covering the period 1957/3-1967/4 (with 7 series each with 42 observations), as published in Economic Trends, with information about consumers' expenditure on goods and services, Investment, inventory investment, imports of goods and services, gross domestic product, and personal disposable income. Prothero and Wallis (1976) fitted several models to each series and compared their performance with a multivariate model.

Usage

```
data("EconomicUK")
```

Format

A data frame with 42 observations on the following 8 variables.

`Cd` consumers' expenditure on durable goods

`Cn` consumers' expenditure on all other goods and services

`I` investment (gross domestic fixed capital formation)

`Iv` inventory investment (value of physical increase in stocks and work in progress)

`M` imports of goods and services

`Y` gross domestic product

`Yd` personal disposable income

`year` year with attributed number associated to quarterly period

Source

The data are quarterly, seasonally unadjusted in 1958 prices, covering the period 1957/3-1967/4 (42 observations), as published in Economic Trends.

References

David L. Prothero and Kenneth F. Wallis (1976). "Modelling macroeconomic time series (with discussion)", Journal of the Royal Statistical Society, A, Vol.139, Part 4, pp.468-500.

`GetResiduals` *Extract Residuals from ARIMA, VAR, or any Simulated Fitted Time Series Model*

Description

This utility function is useful to use in the portmanteau functions, `BoxPierce`, `MahdiMcLeod`, `Hosking`, `LiMcLeod`, `LjungBox`, and `portest`. `GetResiduals()` function takes a fitted time-series object with class `"ar"`, `"arima0"`, `"Arima"`, (`"ARIMA forecast ARIMA Arima"`), `"lm"`, (`"glm" "lm"`), `"varest"`, or `"list"`. and returns the residuals and the order from the fitted object.

Usage

```
GetResiduals(obj)
```

Arguments

`obj`	a fitted time-series model with class `"ar"`, `"arima0"`, `"Arima"`, (`"ARIMA forecast ARIMA Arima"`), `"lm"`, (`"glm" "lm"`), `"varest"`, or `"list"`.

Value

List of order of fitted time series model and residuals from this model.

Author(s)

Esam Mahdi and A.I. McLeod.

See Also

ar, ar.ols, ar.burg, ar.yw, ar.mle, arima0, arima, Arima, auto.arima, lm, glm, VAR, BoxPierce, LjungBox, MahdiMcLeod, Hosking, LiMcLeod.

Examples

```
fit <- arima(Nile, c(1, 0, 1))
GetResiduals(fit)
```

GNPDEF *GNP Deflator for U.S. Inflation Data from January 01, 1947 to April 01, 2010.*

Description

GNP deflator for U.S. inflation data from 1947-01-01 to 2010-04-01.

Usage

```
data(GNPDEF)
```

Format

A data frame with 254 observations on the following 2 variables.

`time` time

`GNPDEF` a numeric vector denotes the GNP deflator

References

Bollerslev, T. (1986). "Generalized autoregressive conditional heteroskedasticity". Journal of Econometrics, 31(3), 307-327.

Examples

```
plot(ts(GNPDEF[,2]))
```

Hosking *The Modified Multivariate Portmanteau Test, Hosking (1980)*

Description

The modified multivariate portmanteau test suggested by Hosking (1980).

Usage

```
Hosking(obj,lags=seq(5,30,5),fitdf=0,sqrd.res=FALSE)
```

Arguments

obj	a univariate or multivariate series with class "numeric", "matrix", "ts", or ("mts" "ts"). It can be also an object of fitted time-series model with class "ar", "arima0", "Arima", ("ARIMA forecast ARIMA Arima"), "lm", ("glm" "lm"), or "varest". obj may also an object with class "list" (see details and following examples).
lags	vector of lag auto-cross correlation coefficients used for Hosking test.
fitdf	Default is zero for testing the randomness of a given sequence with class "numeric", "matrix", "ts", or ("mts" "ts"). In general fitdf equals to the number of estimated parameters in the fitted model. If obj is an object with class "ar", "arima0", "Arima", "varest", ("ARIMA forecast ARIMA Arima"), or "list" then no need to enter the value of fitdf as it will be automatically determined. For obj with other classes, the fitdf is needed for degrees of freedom of asymptotic chi-square distribution.
sqrd.res	if TRUE then apply the test on the squared values. This checks for Autoregressive Conditional Heteroscedastic, ARCH, effects. When sqrd.res = FALSE, then apply the test on the usual residuals.

Details

However the portmanteau test statistic can be applied directly on the output objects from the built in R functions ar(), ar.ols(), ar.burg(), ar.yw(), ar.mle(), arima(), arim0(), Arima(), auto.arima(), lm(), glm(), and VAR(), it works with output objects from any fitted model. In this case, users should write their own function to fit any model they want, where they may use the built in R functions garch(), garchFit(), fracdiff(), tar(), etc. The object obj represents the output of this function. This output must be a list with at least two outcomes: the fitted residual and the fitdf of the fitted model (list(res = ..., fitdf = ...)). See the following example with the function FitModel().

Value

The multivariate test statistic suggested by Hosking (1980) and its associated p-values for different lags based on the asymptotic chi-square distribution with k^2(lags-fitdf) degrees of freedom.

Author(s)

Esam Mahdi and A.I. McLeod.

References

Hosking, J. R. M. (1980). "The Multivariate Portmanteau Statistic". Journal of American Statistical Association, 75, 602-608.

See Also

Box.test, BoxPierce, LjungBox, MahdiMcLeod, LiMcLeod, portest, GetResiduals.

Examples

```
x <- rnorm(100)
Hosking(x)                              ## univariate test
x <- cbind(rnorm(100),rnorm(100))
Hosking(x)                              ## multivariate test
##
##
## Quarterly, west German investment, income, and consumption from 1960 Q1  to 1982 Q4
data(WestGerman)
DiffData <- matrix(numeric(3 * 91), ncol = 3)
  for (i in 1:3)
    DiffData[, i] <- diff(log(WestGerman[, i]), lag = 1)
fit <- ar.ols(DiffData, intercept = TRUE, order.max = 2)
lags <- c(5,10)
## Apply the test statistic on the fitted model (fitdf will be automatically applied)
Hosking(fit,lags,fitdf = 2)                        ## Correct (no need to specify fitdf)
Hosking(fit,lags)                                    ## Correct
## Apply the test statistic on the residuals
res <- ts((fit$resid)[-(1:2), ])
Hosking(res,lags,fitdf = 2)                          ## Correct
Hosking(res,lags)                                    ## Wrong (fitdf is needed!)
##
##
## Write a function to fit a model: Apply portmanteau test on fitted obj with class "list"
FitModel <- function(data){
    fit <- ar.ols(data, intercept = TRUE, order.max = 2)
    fitdf <- 2
    res <- res <- ts((fit$resid)[-(1:2), ])
 list(res=res,fitdf=fitdf)
}
data(WestGerman)
DiffData <- matrix(numeric(3 * 91), ncol = 3)
  for (i in 1:3)
    DiffData[, i] <- diff(log(WestGerman[, i]), lag = 1)
Fit <- FitModel(DiffData)
Hosking(Fit)
```

IbmSp500 *Monthly Returns of IBM and S&P 500 Index*

Description

The monthly returns of IBM stock and the S&P 500 index from January 1926 to December 2008. This data has been discussed by Tsay (2010, Chapter 8).

Usage

```
data(IbmSp500)
```

Format

A data frame with 996 observations on the following 3 variables.

`date` a numeric vector

`ibm` a numeric vector

`sp` a numeric vector

Source

http://faculty.chicagobooth.edu/ruey.tsay/teaching/fts3/

References

Tsay, R. S. (2010). "Analysis of Financial Time Series". Wiley, New York, 3rd edition.

Examples

```
data(IbmSp500)
plot(IbmSp500)
acf(IbmSp500)
```

ImpulseVMA *The Impulse Response Function in the Infinite MA or VMA Representation*

Description

The impulse coefficients are computed.

Usage

```
ImpulseVMA(ar=NULL,ma=NULL,trunc.lag=NULL)
```

Arguments

- `ar` — a numeric or an array of AR or an array of VAR parameters with order p.
- `ma` — a numeric or an array of MA or an array of VMA parameters with order q.
- `trunc.lag` — truncation lag is used to truncate the infinite MA or VMA Process. IF it is NULL, then the default `trunc.lag` = $p + q$.

Value

The impulse response coefficients of order `trunc.lag+1` obtained by converting the ARMA(p, q) or VARMA(p, q) process to infinite MA or VMA process, respectively.

Author(s)

Esam Mahdi and A.I. McLeod.

References

Lutkepohl, H. (2005). "New introduction to multiple time series analysis". Springer-Verlag, New York.

Reinsel, G. C. (1997). "Elements of Multivariate Time Series Analysis". Springer-Verlag, 2nd edition.

See Also

ARMAtoMA, varima.sim, vma.sim, InvertQ

Examples

```
######################################################################
### Impulse response coefficients from AR(1,1) to infinite MA process.
### The infinite process is truncated at lag 20
###
k <- 1
trunc.lag <- 20
ar <- 0.7
ma <- array(-0.9,dim=c(k,k,1))
ImpulseVMA(ar,ma,trunc.lag)
######################################################################
### Impulse response coefficients from VAR(2) to infinite VMA process
### The infinite process is truncated at default lag value = p+q
###
k <- 2
ar <- array(c(0.5,0.4,0.1,0.5,0,0.3,0,0),dim=c(k,k,2))
ma <- NULL
ImpulseVMA(ar,ma)
######################################################################
### Impulse response coefficients from VARMA(2,1) to infinite VMA process
### The infinite process is truncated at lag 50
###
k <- 2
```

```
ar <- array(c(0.5,0.4,0.1,0.5,0,0.25,0,0),dim=c(k,k,2))
ma <- array(c(0.6,0,0.2,0.3),dim=c(k,k,1))
ImpulseVMA(ar,ma,trunc.lag=50)
```

InvertQ *Check Stationary and Invertibility of ARMA or VARMA Models*

Description

Utility function checks whether ARMA or VARMA model satisfies the stationary or/and the invertibility conditions.

Usage

```
InvertQ(coef)
```

Arguments

coef	a numeric, matrix, or array.

Details

It should be noted that, the AR(p) or VAR(p) model can always be expressed as a kp-dimensional AR(1) or VAR(1), and the MA(q) or VMA(q) model can always be expressed as a kq-dimensional MA(1) or VMA(1). For this reason, we can use this fact when we need to find the explicit solutions of AR(p) or VAR(p) models or MA(q) or VMA(q) models as the AR(1) or VAR(1) or the MA(1) or VMA(1) models can be characterized with simple intuitive formulas.

Value

A warning message only if the model is not stationary or/and not invertible.

Author(s)

Esam Mahdi and A.I. McLeod.

References

Lutkepohl, H. (2005). "New introduction to multiple time series analysis". Springer-Verlag, New York.

Reinsel, G. C. (1997). "Elements of Multivariate Time Series Analysis". Springer-Verlag, 2nd edition.

See Also

varima.sim, vma.sim, ImpulseVMA

Examples

```
##############################################################
### Check Stationary
phi <- array(c(0.5,0.4,0.1,0.5,0,0.3,0,0),dim=c(2,2,2))
InvertQ(phi)
### Check Invertibility
theta <- array(c(0.5,0.4,0.1,0.5,0,0.3,0,0),dim=c(2,2,2))
InvertQ(theta)
```

LiMcLeod *The Modified Multivariate Portmanteau Test, Li-McLeod (1981)*

Description

The modified multivariate portmanteau test suggested by Li and McLeod (1981).

Usage

```
LiMcLeod(obj,lags=seq(5,30,5),fitdf=0,sqrd.res=FALSE)
```

Arguments

obj	a univariate or multivariate series with class "numeric", "matrix", "ts", or ("mts" "ts"). It can be also an object of fitted time-series model with class "ar", "arima0", "Arima", ("ARIMA forecast ARIMA Arima"), "lm", ("glm" "lm"), or "varest". obj may also an object with class "list" (see details and following examples).
lags	vector of lag auto-cross correlation coefficients used for Hosking test.
fitdf	Default is zero for testing the randomness of a given sequence with class "numeric", "matrix", "ts", or ("mts" "ts"). In general fitdf equals to the number of estimated parameters in the fitted model. If obj is an object with class "ar", "arima0", "Arima", "varest", ("ARIMA forecast ARIMA Arima"), or "list" then no need to enter the value of fitdf as it will be automatically determined. For obj with other classes, the fitdf is needed for degrees of freedom of asymptotic chi-square distribution.
sqrd.res	if TRUE then apply the test on the squared values. This checks for Autoregressive Conditional Heteroscedastic, ARCH, effects. When sqrd.res = FALSE, then apply the test on the usual residuals.

Details

However the portmanteau test statistic can be applied directly on the output objects from the built in R functions ar(), ar.ols(), ar.burg(), ar.yw(), ar.mle(), arima(), arim0(), Arima(), auto.arima(), lm(), glm(), and VAR(), it works with output objects from any fitted model. In this case, users should write their own function to fit any model they want, where they may use the built in R functions garch(), garchFit(), fracdiff(), tar(), etc. The object obj represents the output of this function. This output must be a list with at least two outcomes: the fitted residual and the fitdf of the fitted model (`list(res = ..., fitdf = ...)`). See the following example with the function `FitModel()`.

Value

The multivariate test statistic suggested by Li and McLeod (1981) and its corresponding p-values for different lags based on the asymptotic chi-square distribution with `k^2(lags-fitdf)` degrees of freedom.

Author(s)

Esam Mahdi and A.I. McLeod.

References

Li, W. K. and McLeod, A. I. (1981). "Distribution of The Residual Autocorrelations in Multivariate ARMA Time Series Models". Journal of The Royal Statistical Society, Series B, 43, 231-239.

See Also

`Box.test`, `BoxPierce`, `LjungBox`, `MahdiMcLeod`, `Hosking`, `portest`, `GetResiduals`.

Examples

```
x <- rnorm(100)
LiMcLeod(x)                              ## univariate test
x <- cbind(rnorm(100),rnorm(100))
LiMcLeod(x)                              ## multivariate test
##
##
## Monthly log stock returns of Intel corporation data: Test for ARCH Effects
monthintel <- as.ts(monthintel)
LjungBox(monthintel)                         ## Usual test
LjungBox(monthintel,sqrd.res=TRUE)  ## Test for ARCH effects
##
##
## Quarterly, west German investment, income, and consumption from 1960 Q1 to 1982 Q4
data(WestGerman)
DiffData <- matrix(numeric(3 * 91), ncol = 3)
  for (i in 1:3)
    DiffData[, i] <- diff(log(WestGerman[, i]), lag = 1)
fit <- ar.ols(DiffData, intercept = TRUE, order.max = 2)
lags <- c(5,10)
## Apply the test statistic on the fitted model (fitdf will be automatically applied)
LiMcLeod(fit,lags,fitdf = 2)                    ## Correct (no need to specify fitdf)
LiMcLeod(fit,lags)                                 ## Correct
## Apply the test statistic on the residuals
res <- ts((fit$resid)[-(1:2), ])
LiMcLeod(res,lags,fitdf = 2)                       ## Correct
LiMcLeod(res,lags)                                 ## Wrong (fitdf is needed!)
##
```

```
##
## Write a function to fit a model: Apply portmanteau test on fitted obj with class "list"
FitModel <- function(data){
    fit <- ar.ols(data, intercept = TRUE, order.max = 2)
    fitdf <- 2
    res <- res <- ts((fit$resid)[-(1:2), ])
 list(res=res,fitdf=fitdf)
}
data(WestGerman)
DiffData <- matrix(numeric(3 * 91), ncol = 3)
  for (i in 1:3)
    DiffData[, i] <- diff(log(WestGerman[, i]), lag = 1)
Fit <- FitModel(DiffData)
LiMcLeod(Fit)
```

LjungBox *Ljung and Box Portmanteau Test*

Description

The Ljung-Box (1978) modified portmanteau test. In the multivariate time series, this test statistic is asymptotically equal to Hosking.

Usage

```
LjungBox(obj,lags=seq(5,30,5),fitdf=0,sqrd.res=FALSE)
```

Arguments

obj	a univariate or multivariate series with class "numeric", "matrix", "ts", or ("mts" "ts"). It can be also an object of fitted time-series model with class "ar", "arima0", "Arima", ("ARIMA forecast ARIMA Arima"), "lm", ("glm" "lm"), or "varest". obj may also an object with class "list" (see details and following examples).
lags	vector of lag auto-cross correlation coefficients used for Hosking test.
fitdf	Default is zero for testing the randomness of a given sequence with class "numeric", "matrix", "ts", or ("mts" "ts"). In general fitdf equals to the number of estimated parameters in the fitted model. If obj is an object with class "ar", "arima0", "Arima", "varest", ("ARIMA forecast ARIMA Arima"), or "list" then no need to enter the value of fitdf as it will be automatically determined. For obj with other classes, the fitdf is needed for degrees of freedom of asymptotic chi-square distribution.
sqrd.res	if TRUE then apply the test on the squared values. This checks for Autoregressive Conditional Heteroscedastic, ARCH, effects. When sqrd.res = FALSE, then apply the test on the usual residuals.

Details

However the portmanteau test statistic can be applied directly on the output objects from the built in R functions `ar()`, `ar.ols()`, `ar.burg()`, `ar.yw()`, `ar.mle()`, `arima()`, `arim0()`, `Arima()`, `auto.arima()`, `lm()`, `glm()`, and `VAR()`, it works with output objects from any fitted model. In this case, users should write their own function to fit any model they want, where they may use the built in R functions `garch()`, `garchFit()`, `fracdiff()`, `tar()`, etc. The object `obj` represents the output of this function. This output must be a list with at least two outcomes: the fitted residual and the fitdf of the fitted model (`list(res = ..., fitdf = ...)`). See the following example with the function `FitModel()`.

Note: In `stats` R, the function `Box.test` was built to compute the Box and Pierce (1970) and Ljung and Box (1978) test statistics only in the univariate case where we can not use more than one single lag value at a time. The functions `BoxPierce` and `LjungBox` are more accurate than `Box.test` function and can be used in the univariate or multivariate time series at vector of different lag values as well as they can be applied on an output object from a fitted model described in the description of the function `BoxPierce`.

Value

The Ljung and Box test statistic with the associated p-values for different lags based on the asymptotic chi-square distribution with `k^2(lags-fitdf)` degrees of freedom.

Author(s)

Esam Mahdi and A.I. McLeod.

References

Ljung, G.M. and Box, G.E.P (1978). "On a Measure of Lack of Fit in Time Series Models". Biometrika, 65, 297-303.

See Also

`Box.test`, `BoxPierce`, `MahdiMcLeod`, `Hosking`, `MahdiMcLeod`, `portest`, `GetResiduals`.

Examples

```
x <- rnorm(100)
LjungBox(x)                              ## univariate test
x <- cbind(rnorm(100),rnorm(100))
LjungBox(x)                              ## multivariate test
##
##
## Annual flow of the river Nile at Aswan - 1871 to 1970
fit <- arima(Nile, c(1, 0, 1))
lags <- c(5, 10, 20)
## Apply the univariate test statistic on the fitted model
LjungBox(fit, lags)            ## Correct (no need to specify fitdf)
LjungBox(fit, lags, fitdf = 2) ## Correct
## Apply the test statistic on the residuals and set fitdf = 2
res <- resid(fit)
```

```
LjungBox(res, lags)             ## Wrong (fitdf is needed!)
LjungBox(res, lags, fitdf = 2)  ## Correct
##
##
## Quarterly, west German investment, income, and consumption from 1960 Q1 to 1982 Q4
data(WestGerman)
DiffData <- matrix(numeric(3 * 91), ncol = 3)
  for (i in 1:3)
    DiffData[, i] <- diff(log(WestGerman[, i]), lag = 1)
fit <- ar.ols(DiffData, intercept = TRUE, order.max = 2)
lags <- c(5,10)
## Apply the test statistic on the fitted model
LjungBox(fit,lags)                ## Correct (no need to specify fitdf)
## Apply the test statistic on the residuals where fitdf = 2
res <- ts((fit$resid)[-(1:2), ])
LjungBox(res,lags)                ## Wrong (fitdf is needed!)
LjungBox(res,lags,fitdf = 2)      ## Correct
##
##
## Monthly log stock returns of Intel corporation data: Test for ARCH Effects
monthintel <- as.ts(monthintel)
LjungBox(monthintel)                         ## Usual test
LjungBox(monthintel,sqrd.res=TRUE)  ## Test for ARCH effects
##
#### Write a function to fit a model: Apply portmanteau test on fitted obj with class "list"
## Example
FitModel <- function(data){
    fit <- ar.ols(data, intercept = TRUE, order.max = 2)
    fitdf <- 2
    res <- res <- ts((fit$resid)[-(1:2), ])
 list(res=res,fitdf=fitdf)
}
data(WestGerman)
DiffData <- matrix(numeric(3 * 91), ncol = 3)
  for (i in 1:3)
    DiffData[, i] <- diff(log(WestGerman[, i]), lag = 1)
Fit <- FitModel(DiffData)
LjungBox(Fit)
```

MahdiMcLeod *Generalized Variance Portmanteau Test*

Description

New generalized variance portmanteau test based on the determinant of the Hosking's autocorrelation block Toeplitz matrix with fitdf $m + 1$ given in the function ToeplitzBlock, where m represents the fitdf of the block matrix. Originally, the generalized variance portmanteau test, MahdiMcLeod, for univariate time series was derived by Pena and Rodriguez (2002) based on the gamma distribution. Lin and McLeod (2006) proposed the Monte-Carlo version of this test and

Mahdi and McLeod (2012) extended both methods to the multivariate case. Simulation results suggest that the Monte-Carlo version of `MahdiMcLeod` statistic is more accurate and powerful than its competitors proposed by Box and Pierce (1970), Ljung and Box (1978), and Pena and Rodriguez (2002, 2006) in the univariate time series and Hosking (1980) and Li and McLeod (1981) in the multivariate time series.

Usage

```
MahdiMcLeod(obj,lags=seq(5,30,5),fitdf=0,sqrd.res=FALSE)
```

Arguments

`obj`	a univariate or multivariate series with class `"numeric"`, `"matrix"`, `"ts"`, or (`"mts"` `"ts"`). It can be also an object of fitted time-series model with class `"ar"`, `"arima0"`, `"Arima"`, (`"ARIMA forecast ARIMA Arima"`), `"lm"`, (`"glm"` `"lm"`), or `"varest"`. `obj` may also an object with class `"list"` (see details and following examples).
`lags`	vector of lag auto-cross correlation coefficients used for `Hosking` test.
`fitdf`	Default is zero for testing the randomness of a given sequence with class `"numeric"`, `"matrix"`, `"ts"`, or (`"mts"` `"ts"`). In general `fitdf` equals to the number of estimated parameters in the fitted model. If `obj` is an object with class `"ar"`, `"arima0"`, `"Arima"`, `"varest"`, (`"ARIMA forecast ARIMA Arima"`), or `"list"` then no need to enter the value of `fitdf` as it will be automatically determined. For `obj` with other classes, the `fitdf` is needed for degrees of freedom of asymptotic chi-square distribution.
`sqrd.res`	if `TRUE` then apply the test on the squared values. This checks for Autoregressive Conditional Heteroscedastic, `ARCH`, effects. When `sqrd.res = FALSE`, then apply the test on the usual residuals.

Details

However the portmanteau test statistic can be applied directly on the output objects from the built in `R` functions `ar()`, `ar.ols()`, `ar.burg()`, `ar.yw()`, `ar.mle()`, `arima()`, `arim0()`, `Arima()`, `auto.arima()`, `lm()`, `glm()`, and `VAR()`, it works with output objects from any fitted model. In this case, users should write their own function to fit any model they want, where they may use the built in `R` functions `garch()`, `garchFit()`, `fracdiff()`, `tar()`, etc. The object `obj` represents the output of this function. This output must be a list with at least two outcomes: the fitted residual and the fitdf of the fitted model (`list(res = ..., fitdf = ...)`). See the following example with the function `FitModel()`.

Value

The generalized variance portmanteau test statistic and its associated p-values for different lags based on asymptotic chi-square as given in Mahdi and McLeod (2012).

Author(s)

Esam Mahdi and A.I. McLeod.

References

Hosking, J. R. M. (1980). "The Multivariate Portmanteau Statistic". Journal of American Statistical Association, 75, 602-608.

Li, W. K. and McLeod, A. I. (1981). "Distribution of The Residual Autocorrelations in Multivariate ARMA Time Series Models". Journal of The Royal Statistical Society, Series B, 43, 231-239.

Lin, J.-W. and McLeod, A.I. (2006). "Improved Generalized Variance Portmanteau Test". Computational Statistics and Data Analysis 51, 1731-1738.

Mahdi, E. and McLeod, A.I. (2012). "Improved Multivariate Portmanteau Test". Journal of Time Series Analysis, 33(2), 211-222.

Pena, D. and Rodriguez, J. (2002). "A Powerful Portmanteau Test of Lack of Test for Time Series". Journal of American Statistical Association, 97, 601-610.

Pena, D. and Rodriguez, J. (2006). "The log of the determinant of the autocorrelation matrix for testing goodness of fit in time series". Journal of Statistical Planning and Inference, 136, 2706-2718.

See Also

acf, ToeplitzBlock, Box.test, BoxPierce, LjungBox, Hosking, LiMcLeod, portest, GetResiduals.

Examples

```
x <- rnorm(100)
MahdiMcLeod(x)                              ## univariate test
x <- cbind(rnorm(100),rnorm(100))
MahdiMcLeod(x)                              ## multivariate test
##
##
## Annual flow of the river Nile at Aswan - 1871 to 1970
fit <- arima(Nile, c(1, 0, 1))
lags <- c(5, 10, 20)
## Apply the univariate test statistic on the fitted model
MahdiMcLeod(fit, lags)            ## Correct (no need to specify fitdf)
MahdiMcLeod(fit, lags, fitdf = 2) ## Correct
## Apply the test statistic on the residuals and set fitdf = 2
res <- resid(fit)
MahdiMcLeod(res, lags)             ## Wrong (fitdf is needed!)
MahdiMcLeod(res, lags, fitdf = 2)  ## Correct
##
##
## Quarterly, west German investment, income, and consumption from 1960 Q1 to 1982 Q4
data(WestGerman)
DiffData <- matrix(numeric(3 * 91), ncol = 3)
  for (i in 1:3)
    DiffData[, i] <- diff(log(WestGerman[, i]), lag = 1)
fit <- ar.ols(DiffData, intercept = TRUE, order.max = 2)
lags <- c(5,10)
## Apply the test statistic on the fitted model
MahdiMcLeod(fit,lags)                ## Correct (no need to specify fitdf)
## Apply the test statistic on the residuals where fitdf = 2
res <- ts((fit$resid)[-(1:2), ])
```

```
MahdiMcLeod(res,lags)                ## Wrong (fitdf is needed!)
MahdiMcLeod(res,lags,fitdf = 2)      ## Correct
##
##
## Monthly log stock returns of Intel corporation data: Test for ARCH Effects
monthintel <- as.ts(monthintel)
MahdiMcLeod(monthintel)                          ## Usual test
MahdiMcLeod(monthintel,sqrd.res=TRUE)  ## Test for ARCH effects
##
#### Write a function to fit a model: Apply portmanteau test on fitted obj with class "list"
## Example
FitModel <- function(data){
    fit <- ar.ols(data, intercept = TRUE, order.max = 2)
    fitdf <- 2
    res <- res <- ts((fit$resid)[-(1:2), ])
 list(res=res,fitdf=fitdf)
}
data(WestGerman)
DiffData <- matrix(numeric(3 * 91), ncol = 3)
  for (i in 1:3)
    DiffData[, i] <- diff(log(WestGerman[, i]), lag = 1)
Fit <- FitModel(DiffData)
MahdiMcLeod(Fit)
```

`monthintel` *The Monthly Log Stock Returns of Intel Corporation from January 1973 to December 2003*

Description

The monthly log stock returns of Intel Corporation from January 1973 to December 2003. This data has been discussed by Tsay (2005, p.99-102). There are 372 data values.

Usage

```
data(monthintel)
```

References

Tsay, R. S. (2005). "Analysis of Financial Time Series". Wiley, New York, 2nd edition.

Examples

```
acf(monthintel)
```

portest *Portmanteau Test Statistics*

Description

Univariate or multivariate portmanteau test statistics of BoxPierce, MahdiMcLeod, Hosking, LiMcLeod, LjungBox, and possibly any other test statistic using Monte-Carlo techniques or asymptotic distributions.

Usage

```
portest(obj,lags=seq(5,30,5),test=c("MahdiMcLeod","BoxPierce","LjungBox",
     "Hosking","LiMcLeod","other"),fn=NULL,sqrd.res=FALSE,MonteCarlo=TRUE,
        innov.dist=c("Gaussian","t","bootstrap"), ncores=1,nrep=1000,
     model=list(sim.model=NULL,fit.model=NULL),pkg.name=NULL,set.seed=123,...)
```

Arguments

obj	if obj is an object of class "ar", "arima0", "Arima", ("ARIMA forecast ARIMA Arima"), "lm", ("glm" "lm"), "varest", or "list" then a portmanteau goodness-of-fit test is done on the fitted model. Otherwise, for obj with class "ts", "numeric", "matrix", or ("mts" "ts"), a test of randomness is done.
lags	vector of lag values is used for portmanteau test.
test	portmanteau test statistic type.
sqrd.res	as described in BoxPierce, MahdiMcLeod, Hosking, LiMcLeod, and LjungBox.
fn	a function calculates the test statistic that is associated with test = "other". For example, fn can be a function returns the generalized Durbin-Watson test statistic values calculated at different lags. This function has at least two inputs: obj and lags, where obj and lags are described as above.
MonteCarlo	if TRUE then apply the Monte-Carlo version of portmanteau statistic. Otherwise, apply the asymptotic distribution.
innov.dist	distribution to generate univariate or multivariate innovation process. This could be Gaussian, t, or bootstrap using resampled errors rather than distributed errors. Default is Gaussian.
ncores	number of cores needed to use in parallel calculations. Default is a single CPU.
nrep	number of replications needed for Monte-Carlo test.
model	additional argument defined as a list with two specified functions, sim.model and fit.model. This argument is needed when the class of obj is "list" (see details and following example).
pkg.name	the name of the required library to be loaded if the Monte-Carlo significance test is used with an object obj with class "list".
set.seed	set.seed is initialized. Default seed is 123, but users can use any seed they wish.

`...` arguments to be passed to methods, such as `dft` degrees of freedom needed to generate innovations with univariate/multivariate series with t-distribution innovations, or `trunc.lag` used in `varima.sim` function, or `fitdf` and `season` as described in `BoxPierce`, `LjungBox`, `Hosking`, `LiMcLeod` and `MahdiMcLeod`.

Details

The portmanteau test statistics, `MahdiMcLeod`, `BoxPierce`, `LjungBox`, `Hosking`, and `LiMcLeod` are implemented based on the Monte-Carlo techniques and the approximation asymptotic distributions as described in Mahdi and McLeod (2012). Any other possible test statistic is also implemented in this function by selecting the argument `test = "other"` and providing the test statistic as a function passing the argument `fn`. The null hypothesis assuming that the fitted model is an adequate model and the residuals behave like white noise series. This function can be used for testing the adequacy in the nonseasonal fitted time series models. this function can be used to check for randomness as well as to check for `ARCH-GARCH` effects. Any other fitted model, for example, threashold autoregression model, may also be tested for adequacy. In this case, two functions, `sim.model()` and `fit.model()`, must be provided via the argument `func`. The object `obj` is the output of the fitted model coded in the function `fit.model` and it is a `"list"` with at least `res`, the residuals from the fitted model in `fit.model()`, and `fitdf`, the fitdf of this fitted model. The output from the function `sim.model()` is a simulated univariate or multivariate series from the fitted model obtained from the function `fit.model()`. The argument `pkg.name` represents the name of the `R` package where the fitted model build in (see the last given example). The parallel computing using the `portes` package proposed by Gonzalo Vera, Ritsert Jansen, and Remo Suppi (2008) will run if one decide to choose the argument `MonteCarlo=TRUE` provided that `ncores` equals to a positive integer more than 1.

Value

The portmanteau test statistic with the associated p-values for different lag values. When the argument `MonteCarlo` is set to be `FALSE` then the degrees of freedom will be an additional output.

Author(s)

Esam Mahdi and A.I. McLeod.

References

Box, G.E.P. and Pierce, D.A. (1970). "Distribution of Residual Autocorrelation in Autoregressive-Integrated Moving Average Time Series Models". Journal of American Statistical Association, 65, 1509-1526.

Fox, J and Weisberg, S and Adler, D and Bates, D and Baud-Bovy, G and Ellison, S and Firth, D and Friendly, M and Gorjanc, G and Graves, S and Heiberger, R and Laboissiere, R and Monette, G and Murdoch, D and Nilsson, H and Ogle, D and Ripley, B and Venables, W and Zeileis, A and R-Core (2019). car: Companion to Applied Regression. R package version 3.0-3, https://CRAN.R-project.org/package=car.

Fraley C, Leisch F, Maechler M, Reisen V, Lemonte A (2012). fracdiff: Fractionally differenced ARIMA aka ARFIMA(p,d,q) models. R package version 1.4-2, https://CRAN.R-project.org/package=fracdiff.

Hosking, J. R. M. (1980). "The Multivariate Portmanteau Statistic". Journal of American Statistical Association, 75, 602-608.

John Haslett and Adrian E. Raftery (1989). "Space-time Modelling with Long-memory Dependence: Assessing Ireland's Wind Power Resource (with Discussion)". Applied Statistics, 38, 1-50.

Li, W. K. and McLeod, A. I. (1981). "Distribution of The Residual Autocorrelations in Multivariate ARMA Time Series Models". Journal of The Royal Statistical Society, Series B, 43, 231-239.

Lin, J.-W. and McLeod, A.I. (2006). "Improved Generalized Variance Portmanteau Test". Computational Statistics and Data Analysis 51, 1731-1738.

Lin, J.-W. and McLeod, A.I. (2008). "Portmanteau Tests for ARMA Models with Infinite Variance". Journal of Time Series Analysis, 29, 600-617.

Ljung, G.M. and Box, G.E.P (1978). "On a Measure of Lack of Fit in Time Series Models". Biometrika, 65, 297-303.

Mahdi, E. and McLeod, A.I. (2012). "Improved Multivariate Portmanteau Test". Journal of Time Series Analysis, 33(2), 211-222.

McLeod A.I, Li W.K (1983). "Distribution of the Residual Autocorrelation in Multivariate ARMA Time Series Models". Journal of Time Series Analysis, 4, 269-273.

Pena, D. and Rodriguez, J. (2002). "A Powerful Portmanteau Test of Lack of Test for Time Series". Journal of American Statistical Association, 97, 601-610.

Pena, D. and Rodriguez, J. (2006). "The log of the determinant of the autocorrelation matrix for testing goodness of fit in time series". Journal of Statistical Planning and Inference, 136, 2706-2718.

Pfaff B, Stigler M (2018). vars: VAR Modelling. R package version 1.5-3, https://CRAN.R-project.org/package=vars.

Rob J Hyndman with contributions from George Athanasopoulos Slava Razbash DSZZYKCB, Wang E (2019). forecast: Forecasting Functions for Time Series and Linear Models. R package version 8.7, https://CRAN.R-project.org/package=forecast.

Tierney, L., Rossini, A. J., Li, N., and Sevcikova, H. (2018). snow: Simple Network of Workstations. R package version 0.4-3. https://CRAN.R-project.org/package=snow.

Trapletti A, Hornik K, LeBaron B (2019). tseries: Time Series Analysis and Computational Finance. R package version 0.10-47, https://CRAN.R-project.org/package=tseries.

Gonzalo Vera and Ritsert C. Jansen and Remo L. Suppi (2008). R/parallel - speeding up bioinformatics analysis with R. BMC Bioinformatics, 9:390.

Wuertz D, core team members R (2019). fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling. R package version 3042.83.1, https://CRAN.R-project.org/package=fGarch.

See Also

acf, ar, ar.ols, ar.burg, ar.yw, ar.mle, arima0, arima, lm, glm, Box.test, BoxPierce, LjungBox, MahdiMcLeod, LiMcLeod, portest, ToeplitzBlock, GetResiduals, Arima, auto.arima, VAR, fracdiff, garchFit, garch, varima.sim.

Examples

```
## Not run:
################################################################################
####                                                                        ####
####                          Portmanteau Tests                             ####
####                                                                        ####
################################################################################
## Monte-Carlo (MC) and asymptotic tests for randomness series               ##
################################################################################
data("DEXCAUS")
returns <- log(DEXCAUS[-1]/DEXCAUS[-length(DEXCAUS)])
portest(returns)            ## MC using one CPU takes about 24.16 seconds
portest(returns, ncores=4)    ## MC using 4 CPUs takes about 9.51 seconds
portest(returns, MonteCarlo=FALSE)              ## asymptotic MahdiMcLeod
portest(returns,test="LjungBox", MonteCarlo=FALSE) ## asymptotic LjungBox
################################################################################
## Monte-Carlo goodness-of-fit arima test using 4 CPUs                       ##
################################################################################
## arima() function takes about 11.32 seconds
## Example 1
ans1 <- arima(WWWusage,fitdf=c(3,1,0))
portest(ans1, ncores = 4)
#
## arima0() function takes about 11.1 seconds
## Example 2
ans2 <- arima0(WWWusage,fitdf=c(3,1,0))
portest(ans2, ncores = 4)
#
## Arima() or auto.arima() functions from forecast package take about 12.1 seconds
## Example 3
ans3 <- Arima(WWWusage,fitdf=c(3,1,0))
portest(ans3, ncores = 4)
#
## ar() function takes about 7.39 seconds
## Example 4
ans4 <- ar(Nile,fitdf.max=2)
portest(ans4, ncores = 4)
#
## ar() function with your own R code takes about 8.75 seconds
## Example 5
fit.model <- function(data){
    fit <- ar(data,aic = FALSE, fitdf.max=2)
    fitdf <- 2
    res <- ts(fit$resid[-(1:fitdf)])
    phi <- fit$ar
    theta <- NULL
    sigma <- fit$var.pred
    demean <- fit$x.mean
 list(res=res,phi=phi,theta=theta,sigma=sigma,demean=demean)
}
sim.model <- function(parSpec){
    res <- parSpec$res
```

```
    n <- length(res)
    innov <- function(n) ts(stats::rnorm(n, mean = demean, sd = sqrt(sigma)))
    phi <- parSpec$phi
    theta <- parSpec$theta
    sigma <- parSpec$sigma
    demean <- parSpec$demean
   arima.sim(n = n, list(ar = phi, ma = theta), rand.gen=innov)
}
ans5 <- fit.model(Nile)
portest(ans5,ncores=4,model=list(sim.model=sim.model,fit.model=fit.model),pkg.name="stats")
#################################################################################
## Monte-Carlo test for seasonality                                            ##
#################################################################################
## Accidental Deaths in the US 1973 - 1978
seasonal.arima<-Arima(USAccDeaths,fitdf=c(0,1,1),seasonal=list(fitdf= c(0,1,1)))
portest(seasonal.arima,ncores=4,nrep=1000,lags=1:5)
## Quarterly U.K. economic time series from 1957 Q3 to 1967 Q4
cd <- EconomicUK[,1]
cd.fit <- Arima(cd,fitdf=c(0,1,0),seasonal=list(fitdf=c(0,1,1),period=4))
portest(cd.fit, lags = c(5,10),ncores=4)
#################################################################################
## Monte-Carlo test for linear models and time series regression               ##
#################################################################################
## Linear Model
require("car")
fit <- lm(fconvict ~ tfr + partic + degrees + mconvict, data=Hartnagel)
portest(fit,lags=1:3,ncores=4)          ## MC of MahdiMcLeod test
## MC of generalized Durbin-Watson test needs the argument function fn() as follows
fn <- function(obj,lags){
     test.stat <- numeric(length(lags))
       for (i in 1:length(lags))
          test.stat[i] <- -sum(diff(obj,lag=lags[i])^2)/sum(obj^2)
       test.stat
}
portest(fit,lags=1:3,test="other",fn=fn,ncores=4)
detach(package:car)
## Time series regression
fit.arima <- Arima(LakeHuron, fitdf = c(2,0,0), xreg = time(LakeHuron)-1920)
portest(fit.arima,ncores=4)
#################################################################################
## Monte-Carlo goodness-of-fit VAR test - Multivariate series                   ##
#################################################################################
data("IbmSp500")
ibm <- log(IbmSp500[,2]+1)*100
sp500 <- log(IbmSp500[,3]+1)*100
IBMSP500 <- data.frame(cbind(ibm,sp500))
## ar.ols() function takes about 9.11 seconds
ans6 <-  ar.ols(IBMSP500, aic=FALSE, intercept=TRUE, fitdf.max=5)
portest(ans6,nrep=100,test="MahdiMcLeod",ncores=4,innov.dist="t",dft=5)
## VAR() function takes about 11.55 seconds
require("vars")
ans7 <- VAR(IBMSP500, p=5)
portest(ans7,nrep=100,test="MahdiMcLeod",ncores=4,innov.dist="bootstrap")
```

```
portest(ans7,test="Hosking",MonteCarlo=FALSE) ## asymptotic Hosking test
detach(package:vars)
##################################################################################
## Monte-Carlo test for ARCH/GARCH effects using 4 CPUs                         ##
##################################################################################
## It takes about 12.65 seconds
data("monthintel")
returns <- as.ts(monthintel)
lags <- c(5, 10, 20, 40)
portest(returns, lags = lags, ncores = 4, sqrd.res = FALSE)
portest(returns,lags=lags,ncores=4,sqrd.res=TRUE,innov.dist="t",dft=5)

## End(Not run)
```

ToeplitzBlock — *Toeplitz Block Matrix of Hosking (1980) Auto and Cross Correlation Matrices*

Description

Block Toeplitz matrix of order $m + 1$ with $k \times k$ auto-cross correlation matrices. The Hosking (1980) definition of the correlation matrix is used. This is needed for the function MahdiMcLeod.

Usage

```
ToeplitzBlock(res,Maxlag)
```

Arguments

res	residuals, numeric or matrix.
Maxlag	an integer number = m is used to determined the order of the block matrix.

Value

A block Toeplitz matrix of auto and cross correlation matrices using Hosking (1980) definition from lag = 0 to lag = m.

Author(s)

Esam Mahdi and A.I. McLeod.

References

Hosking, J. R. M. (1980). "The Multivariate Portmanteau Statistic". Journal of American Statistical Association, 75, 602-608.

Lin, J.-W. and McLeod, A.I. (2006). "Improved Generalized Variance Portmanteau Test". Computational Statistics and Data Analysis, 51, 1731-1738.

Mahdi, E. and McLeod, A.I. (2011, accepted). "Improved Multivariate Portmanteau Test". Journal of Time Series Analysis. (JTSA - 3192).

See Also

acf, MahdiMcLeod, toeplitz

Examples

```
x <- rnorm(100)
ToeplitzBlock(x,Maxlag=4)          ## Univariate Series
#
y <- cbind(rnorm(100),rnorm(100))
ToeplitzBlock(y,Maxlag=4)          ## Multivariate Series
```

---

varima.sim *Simulate Data From Seasonal/Nonseasonal ARIMA(p,d,q)*(ps,ds,qs)_s or VARIMA(p,d,q)*(ps,ds,qs)_s Models*

Description

Simulate time series from AutoRegressive Integrated Moving Average, `ARIMA(p,d,q)`, or Vector Integrated AutoRegressive Moving Average, `VARIMA(p,d,q)`, where `d` is a nonnegative difference integer in the `ARIMA` case and it is a vector of k differenced components $d_1, ..., d_k$ in the `VARIMA` case. In general, this function can be implemented in simulating univariate or multivariate Seasonal AutoRegressive Integrated Moving Average, `SARIMA(p,d,q)*(ps,ds,qs)_s` and `SVARIMA(p,d,q)*(ps,ds,qs)_s`, where `ps` and `qs` are the orders of the seasonal univariate/multivariate AutoRegressive and Moving Average components respectively. `ds` is a nonnegative difference integer in the `SARIMA` case and it is a vector of k differenced components $ds_1, ..., ds_k$ in the `SVARIMA` case, whereas `s` is the seasonal period. The simulated process may have a deterministic terms, drift constant and time trend, with non-zero mean. The innovations may have finite or infinite variance.

Usage

```
varima.sim(model=list(ar=NULL,ma=NULL,d=NULL,sar=NULL,sma=NULL,D=NULL,period=NULL),
           n,k=1,constant=NA,trend=NA,demean=NA,innov=NULL,
           innov.dist=c("Gaussian","t","bootstrap"),...)
```

Arguments

`model`	a list with univariate/multivariate component `ar` and/or `ma` and/or `sar` and/or `sma` giving the univariate/multivariate `AR` and/or `MA` and/or `SAR` and/or `SMA` coefficients respectively. `period` specifies the seasonal period. For seasonality, default is `NULL` indicates that `period =12`. `d` and `D` are integer or vector representing the order of the usual and seasonal difference. An empty list gives an `ARIMA(0, 0, 0)*(0,0,0)_null` model, that is white noise.
`n`	length of the series.
`k`	number of simulated series. For example, `k=1` is used for univariate series and `k=2` is used for bivariate series.

constant	a numeric vector represents the intercept in the deterministic equation.
trend	a numeric vector represents the slop in the deterministic equation.
demean	a numeric vector represents the mean of the series.
innov	a vector of univariate or multivariate innovation series. This may used as an initial series to genrate innovations with innov.dist = "bootstrap". This argument is irrelevant with the other selections of innov.dist.
innov.dist	distribution to generate univariate or multivariate innovation process. This could be Gaussian, t, or bootstrap using resampled errors rather than distributed errors. Default is Gaussian.
...	arguments to be passed to methods, such as dft degrees of freedom needed to generate innovations with univariate/multivariate series with t-distribution innovations. The argument trunc.lag represents the truncation lag that is used to truncate the infinite MA or VMA Process. IF it is not given, then trunc.lag = min(100, $n/3$). Optionally sigma is the variance of a Gaussian or t white noise series.

Details

This function is used to simulate a univariate/multivariate seasonal/nonseasonal SARIMA or SVARIMA model of order $(p, d, q) \times (ps, ds, qs)_s$

$$\phi(B)\Phi(B^s)d(B)D(B^s)(Z_t) - \mu = a + b \times t + \theta(B)\Theta(B^s)e_t,$$

where $a, b,$ and μ correspond to the arguments constant, trend, and demean respectively. The univariate or multivariate series e_t represents the innovations series given from the argument innov. If innov = NULL then e_t will be generated from a univariate or multivariate normal distribution or t-distribution. $\phi(B)$ and $\theta(B)$ are the VAR and the VMA coefficient matrices respectively and B is the backshift time operator. $\Phi(B^s)$ and $\Theta(B^s)$ are the Vector SAR Vector SMA coefficient matrices respectively. $d(B) = diag[(1 - B)^{d_1}, \ldots, (1 - B)^{d_k}]$ and $D(B^s) = diag[(1 - B^s)^{ds_1}, \ldots, (1 - B^s)^{ds_k}]$ are diagonal matrices. This states that each individual series $Z_i, i = 1, ..., k$ is differenced $d_i ds_i$ times to reduce to a stationary Vector ARMA(p,0,q)*(ps,0,qs)_s series.

Value

Simulated data from SARIMA(p,d,q) or SVARIMA(p,d,q)*(ps,ds,qs)_s process that may have a drift and deterministic time trend terms.

Author(s)

Esam Mahdi and A.I. McLeod.

References

Hipel, K.W. and McLeod, A.I. (2005). "Time Series Modelling of Water Resources and Environmental Systems".

Reinsel, G. C. (1997). "Elements of Multivariate Time Series Analysis". Springer-Verlag, 2nd edition.

See Also

arima.sim, vma.sim, ImpulseVMA, InvertQ

Examples

```
################################################################################
# Simulate white noise series from a Gaussian distribution                     #
################################################################################
set.seed(1234)
Z1 <- varima.sim(n=400)       ## a univariate series
plot(Z1)
Z2 <- varima.sim(n=400,k=2)   ## a bivariate series
plot(Z2)
Z3 <- varima.sim(n=400,k=5)   ## a multivariate series of dimension 5
plot(Z3)
################################################################################
# Simulate MA(1) where innovation series is provided via argument innov        #
################################################################################
set.seed(1234)
n <- 200
ma <-  0.6
Z<-varima.sim(list(ma=ma),n=n,innov=rnorm(n),innov.dist="bootstrap")
plot(Z)
################################################################################
# Simulate seasonal ARIMA(2,1,0)*(0,2,1)_12 process with ar=c(1.3,-0.35),      #
# ma.season = 0.8. Gaussian innovations. The series is truncated at lag 50    #
################################################################################
set.seed(12834)
n <- 100
ar <- c(1.3, -0.35)
ma.season <- 0.8
Z<-varima.sim(list(ar=ar,d=1,sma=ma.season,D=2),n=n,trunc.lag=50)
plot(Z)
acf(Z)
################################################################################
# Simulate seasonal ARMA(0,0,0)*(2,0,0)_4 process with intercept = 0.8         #
# ar.season = c(0.9,-0.9), period = 4, t5-distribution innovations with df = 3 #
################################################################################
set.seed(1234)
n <- 200
ar.season <- c(0.9,-0.9)
Z<-varima.sim(list(sar=ar.season,period=4),n=n,constant=0.8,innov.dist="t",dft=3)
plot(Z)
acf(Z)
arima(Z,order=c(0,0,0),seasonal = list(order = c(2,0,0),period=4))
################################################################################
# Simulate a bivariate white noise series from a multivariate t4-distribution   #
# Then use the nonparametric bootstrap method to generate a seasonal SVARIMA    #
# of order (0,d,0)*(0,0,1)_12 with d = c(1, 0), n= 250, k = 2, and             #
# ma.season=array(c(0.5,0.4,0.1,0.3),dim=c(k,k,1))                            #
################################################################################
set.seed(1234)
```

```
Z1 <- varima.sim(n=250,k=2,innov.dist="t",dft=4)
ma.season=array(c(0.5,0.4,0.1,0.3),dim=c(2,2,1))
Z2 <- varima.sim(list(sma=ma.season,d=c(1,0)),n=250,k=2,
                 innov=Z1,innov.dist="bootstrap")
plot(Z2)
#################################################################################
# Simulate a bivariate VARIMA(2,d,1) process with length 300, where d=(1,2).    #
# ar = array(c(0.5,0.4,0.1,0.5,0,0.3,0,0),dim=c(k,k,2)),                       #
# ma = array(c(0,0.25,0,0), dim=c(k,k,1)).                                    #
# innovations are generated from multivariate normal                            #
# The process have mean zero and no deterministic terms.                        #
# The variance covariance is sigma = matrix(c(1,0.71,0.71,2),2,2).              #
# The series is truncated at default value: trunc.lag=ceiling(100/3)=34         #
#################################################################################
set.seed(1234)
k <- 2
n <- 300
ar <-  array(c(0.5,0.4,0.1,0.5,0,0.3,0,0),dim=c(k,k,2))
ma <-  array(c(0,0.25,0,0),dim=c(k,k,1))
d <- c(1,2)
sigma <- matrix(c(1,0.71,0.71,2),k,k)
Z <- varima.sim(list(ma=ar,ma=ma,d=d),n=n,k=2,sigma=sigma)
plot(Z)
#################################################################################
# Simulate a trivariate Vector SVARMA(1,0,0)*(1,0,0)_12 process with length 300 #
# ar = array(c(0.5,0.4,0.1,0.5,0,0.3,0,0,0.1), dim=c(k,k,1)), where k =3       #
# ar.season = array(c(0,0.25,0,0.5,0.1,0.4,0,0.25,0.6), dim=c(k,k,1)).         #
# innovations are generated from multivariate normal distribution               #
# The process have mean c(10, 0, 12),                                           #
# Drift equation a + b * t, where a = c(2,1,5), and b = c(0.01,0.06,0)          #
# The series is truncated at default value: trunc.lag=ceiling(100/3)=34         #
#################################################################################
set.seed(1234)
k <- 3
n <- 300
ar <-  array(c(0.5,0.4,0.1,0.5,0,0.3,0,0,0.1),dim=c(k,k,1))
ar.season <-  array(c(0,0.25,0,0.5,0.1,0.4,0,0.25,0.6),dim=c(k,k,1))
constant <- c(2,1,5)
trend <- c(0.01,0.06,0)
demean <- c(10,0,12)
Z <- varima.sim(list(ar=ar,sar=ar.season),n=n,k=3,constant=constant,
trend=trend,demean=demean)
plot(Z)
acf(Z)
```

vma.sim *Compute The Vector of Moving Average Model (VMA)*

Description

This utility function is useful to use in the function `varima.sim` and may used to compute the coefficients of moving-average or vector moving-average.

Usage

```
vma.sim(psi, a)
```

Arguments

`psi`	the impulse coefficients.
`a`	innovations

Value

Vector of length n (in the univariate case), or n matrices (in the multivariate case), where n = length(a)-length(Ψ) and $n \times k$ is the dimension of the series.

Author(s)

Esam Mahdi and A.I. McLeod.

References

Hannan, E.J. (1970). "Multiple Time Series". New York: Wiley.

Hipel, K.W. and McLeod, A.I. (2005). "Time Series Modelling of Water Resources and Environmental Systems".

See Also

`convolve`, `varima.sim`, `arima.sim`, `ImpulseVMA`, `InvertQ`

Examples

```
k <- 2
n <- 300
trunc.lag <-  50
ar <-  array(c(0.5,0.4,0.1,0.5),dim=c(k,k,1))
ma <-  array(c(0,0.25,0,0),dim=c(k,k,1))
sigma <- matrix(c(1,0.71,0.71,2),k,k)
p <- ifelse(is.null(ar),0,dim(ar)[3])
q <- ifelse(is.null(ma),0,dim(ma)[3])
r <- max(p, q)
d <- trunc.lag + r
psi <- ImpulseVMA(ar = ar, ma = ma, trunc.lag = trunc.lag)
a <- t(crossprod(chol(sigma),matrix(rnorm(k*d),ncol=d)))
vma.sim(psi = psi, a = a)
```

`WestGerman` *Quarterly, West German Investment, Income, and Consumption: 1960Q1-1982Q4*

Description

Quarterly, seasonally adjusted, West German fixed investment, disposable income, consumption expenditures in billions of DM, 1960Q1-1982Q4.

Usage

```
data(WestGerman)
```

Format

A data frame with 92 observations on the following 3 variables.

`invest` a numeric vector denotes the investment in billions of DM

`income` a numeric vector denotes the income in billions of DM

`cons` a numeric vector denotes the consumption expenditures in billions of DM

Source

Deutsche Bundesbank; http://www.jmulti.de/data_imtsa.html

References

Lutkepohl, H. (2005). "New introduction to multiple time series analysis". Springer-Verlag, New York.

Index