Package ‘nortsTest’

November 9, 2025

Title Assessing Normality of Stationary Process

Version 1.1.3

Date 2025-11-09

Description Despite that several tests for normality in stationary processes have been proposed in the literature, consistent implementations of these tests in programming languages are limited. Seven normality test are implemented. The asymptotic Lobato & Velasco's, asymptotic Epps, Psaradakis and Vávra, Lobato & Velasco's and Epps sieve bootstrap approximations, El bouch et al., and the random projections tests for univariate stationary process. Some other diagnostics such as, unit root test for stationarity, seasonal tests for seasonality, and arch effect test for volatility; are also performed. Additionally, the El bouch test performs normality tests for bivariate time series. The package also offers residual diagnostic for linear time series models developed in several packages.

License GPL-2

Encoding UTF-8

NeedsCompilation no

RoxygenNote 7.3.3

Depends R (>= 3.5.0), methods

Imports forecast, nortest, ggplot2, gridExtra, cowplot, tseries, uroot, MASS, zoo

Suggests ggfortify, testthat (>= 3.0.0)

URL `https://github.com/asael697/nortsTest`

BugReports `https://github.com/asael697/nortsTest/issues`

Config/testthat/edition 3

Author Asael Alonzo Matamoros [aut, cre],
Alicia Nieto-Reyes [aut],
Rob Hyndman [ctb],
Mitchell O'Hara-Wild [ctb],
Trapletti A. [ctb]

Maintainer Asael Alonzo Matamoros <`asael.alonzo@gmail.com`>

Repository CRAN

Date/Publication 2025-11-09 18:50:02 UTC

Contents

`nortsTest-package` *Assessing Normality of a Stationary Process*

Description

Despite several tests for normality in stationary processes being proposed in the literature, consistent implementations in programming languages are limited. This package implements seven normality tests: the asymptotic Lobato and Velasco, asymptotic Epps, Psaradakis & Vávra, Lobato & Velasco's sieve bootstrap approximation, Elbouch et al., Epps sieve bootstrap approximation, and the random projections test for univariate stationary processes. Additional diagnostics such as unit root tests, seasonality tests, and ARCH effect tests are also included. The Elbouch test additionally performs bivariate normality testing. Residual diagnostics for linear time series models from several packages are also available.

Details

Functions provided for univariate normality tests include: `epps.test`, `lobato.test`, `rp.test`, `lobato-bootstrap.test`, `epps-bootstrap.test`, `elbouch.test`, and `varvra.test`. The `elbouch.test` function can perform a bivariate test if a second time series is supplied. Model diagnostics functions include unit root, seasonality, and ARCH effect tests. Visual checks can be done with **ggplot2** and **forecast**.

Author(s)

Maintainer: Asael Alonzo Matamoros <asael.alonzo@gmail.com>

Authors:

- Alicia Nieto-Reyes <alicia.nieto@unican.es>

Other contributors:

- Rob Hyndman <Rob.Hyndman@monash.edu> [contributor]
- Mitchell O'Hara-Wild [contributor]
- Trapletti A. [contributor]

References

Epps, T.W. (1987). Testing that a stationary time series is Gaussian. *The Annals of Statistic*, 15(4), 1683-1698. https://projecteuclid.org/euclid.aos/1176350618

Lobato, I., & Velasco, C. (2004). A simple test of normality in time series. *Journal of Econometric Theory*, 20(4), 671-689. doi:https://doi.org/10.1017/S0266466604204030

Psaradakis, Z. & Vávra, M. (2017). A distance test of normality for a wide class of stationary processes. *Journal of Econometrics and Statistics*, 2, 50-60. doi:https://doi.org/10.1016/j.ecosta.2016.11.005

Nieto-Reyes, A., Cuesta-Albertos, J., & Gamboa, F. (2014). A random-projection based test of Gaussianity for stationary processes. *Computational Statistics & Data Analysis*, 75(C), 124-141.

Hyndman, R. & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 26(3), 1-22. doi:10.18637/jss.v027.i03

Wickham, H. (2008). ggplot2: Elegant Graphics for Data Analysis. *Springer-Verlag New York*.

See Also

Useful links:

- https://github.com/asael697/nortsTest
- Report bugs at https://github.com/asael697/nortsTest/issues

`arch.test` *The ARCH effect test function.*

Description

Performs the Pormanteau Q and Lagrange Multipliers test for homoscedasticity in a univariate stationary process. The null hypothesis (H0), is that the process is homoscedastic.

Usage

```
arch.test(y, arch = c("box","Lm"), alpha = 0.05, lag.max = 2)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`arch`	A character string naming the desired test for checking stationarity. Valid values are `"box"` for the Ljung-Box, and `"Lm"` for the Lagrange Multiplier test. The default value is `"box"` for the Augmented Ljung-Box test.
`alpha`	Level of the test, possible values range from 0.01 to 0.1. By default `alpha = 0.05` is used
`lag.max`	an integer with the number of used lags.

Details

Several different tests are available: Performs Portmanteau Q and Lagrange Multiplier tests for the null hypothesis that the residuals of an ARIMA model are homoscedastic. The ARCH test is based on the fact that if the residuals (defined as `e(t)`) are heteroscedastic, the squared residuals (e^2[t]) are autocorrelated. The first type of test is to examine whether the squares of residuals are a sequence of white noise, which is called the Portmanteau Q test, and similar to the Ljung-Box test on the squared residuals. By default, `alpha = 0.05` is used to select the more likely hypothesis.

Value

A list with class `"h.test"` containing the following components:

`statistic:`	the test statistic.
`parameter:`	the test degrees freedoms.
`p.value:`	the p-value for the test.
`alternative:`	a character string describing the alternative hypothesis.
`method:`	a character string with the test name.
`data.name:`	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros

References

Engle, R. F. (1982). Auto-regressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*. 50(4), 987-1007.

McLeod, A. I. & W. K. Li. (1984). Diagnostic Checking ARMA Time Series Models Using Squared-Residual Auto-correlations. *Journal of Time Series Analysis.* 4, 269-273.

See Also

`normal.test`, `seasonal.test`, `uroot.test`

Examples

```
#  stationary  ar process
y = arima.sim(100,model = list(ar = 0.3))
arch.test(y)
```

---

autoplot.ts — *Automatically create a ggplot for time series objects.*

Description

`autoplot` takes an object of type `ts` or `mts` and creates a ggplot object suitable for usage with `stat_forecast`.

Usage

```
## S3 method for class 'ts'
autoplot(
  object,
  series = NULL,
  xlab = "Time",
  ylab = deparse(substitute(object)),
  main = NULL,
  facets = FALSE,
  colour = TRUE,
  ...
)

## S3 method for class 'numeric'
autoplot(
  object,
  series = NULL,
  xlab = "Time",
  ylab = deparse(substitute(object)),
  main = NULL,
  ...
```

```
)

## S3 method for class 'ts'
fortify(model, data, ...)
```

Arguments

`object`	Object of class “`ts`” or “`mts`”.
`series`	Identifies the time series with a colour, which integrates well with the functionality of geom_forecast.
`xlab`	a string with the plot’s x axis label. By default a NULL value.
`ylab`	a string with the plot’s y axis label. By default a counts" value.
`main`	a string with the plot’s title.
`facets`	If TRUE, multiple time series will be faceted (and unless specified, colour is set to FALSE). If FALSE, each series will be assigned a colour.
`colour`	If TRUE, the time series will be assigned a colour aesthetic.
`...`	Other plotting parameters to affect the plot.
`model`	Object of class “`ts`” to be converted to “`data.frame`”.
`data`	Not used (required for the ‘fortify‘ method).

Details

`fortify.ts` takes a `ts` object and converts it into a data frame (for usage with ggplot2).

Value

None. Function produces a ggplot2 graph.

Author(s)

Mitchell O’Hara-Wild

See Also

`plot.ts`, `fortify`

Examples

```
library(ggplot2)
autoplot(USAccDeaths)

lungDeaths <- cbind(mdeaths, fdeaths)
autoplot(lungDeaths)
autoplot(lungDeaths, facets=TRUE)
```

check_plot.ts *Generic function for a visual check of residuals in time series models.*

Description

Generic function for a visual check of residuals in time series models, these methods are inspired in the `check.residuals` function provided by the `forecast` package.

Usage

```
## S3 method for class 'ts'
check_plot(y, model = " ", ...)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`model`	A string with the model name.
`...`	Other plotting parameters to affect the plot.

Value

A graph object from ggplot2.

Author(s)

Asael Alonzo Matamoros.

See Also

`check_residuals`

Examples

```
y = arima.sim(100,model = list(ar = 0.3))
check_plot(y)
```

check_residuals.ts *Generic functions for checking residuals in time series models*

Description

Generic function for residuals check analysis, these methods are inspired in the `check.residuals` function provided by the `forecast` package.

Usage

```
## S3 method for class 'ts'
check_residuals(
  y,
  normality = "epps",
  unit_root = NULL,
  seasonal = NULL,
  arch = NULL,
  alpha = 0.05,
  plot = FALSE,
  ...
)
```

Arguments

y	Either a time series model,the supported classes are `arima0`, `Arima`, `sarima`, `fGarch`, or a time series (assumed to be residuals).
normality	A character string naming the desired test for checking gaussian distribution. Valid values are `"epps"` for the Epps, `"lobato"` for Lobato and Velasco's,`"vavras"` for the Psaradakis and Vávra, `"rp"` for the random projections, `"jb"` for the Jarque and Bera, `"ad"` for Anderson Darling test, and `"shapiro"` for the Shapiro-Wilk's test. The default value is `"epps"` test.
unit_root	A character string naming the desired unit root test for checking stationarity. Valid values are `"adf"` for the Augmented Dickey-Fuller, `"pp"` for the Phillips-Perron, and `"kpss"` for Kwiatkowski, Phillips, Schmidt, and Shin. The default value is `"adf"` for the Augmented Dickey-Fuller test.
seasonal	A character string naming the desired unit root test for checking seasonality. Valid values are `"ocsb"` for the Osborn, Chui, Smith, and Birchenhall, `"ch"` for the Canova and Hansen, and `"hegy"` for Hylleberg, Engle, Granger, and Yoo. The default value is `"ocsb"` for the Osborn, Chui, Smith, and Birchenhall test.
arch	A character string naming the desired test for checking stationarity. Valid values are `"box"` for the Ljung-Box, and `"Lm"` for the Lagrange Multiplier test. The default value is `"box"` for the Augmented Ljung-Box test.
alpha	Level of the test, possible values range from 0.01 to 0.1. By default `alpha = 0.05` is used
plot	A boolean value. If `TRUE`, will produce produces a time plot of the residuals, the corresponding ACF, and a histogram.

... Other testing parameters

Details

The function performs a residuals analysis, it prints a unit root and seasonal test to check stationarity, and a normality test for checking Gaussian distribution. In addition, if the plot option is `TRUE` a time plot, ACF, and histogram of the series are presented.

Value

The function does not return any value

Author(s)

Asael Alonzo Matamoros

References

Dickey, D. & Fuller, W. (1979). Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *Journal of the American Statistical Association*. 74, 427-431.

Epps, T.W. (1987). Testing that a stationary time series is Gaussian. *The Annals of Statistic*. 15(4), 1683-1698.http://www.jstor.org/stable/2336512. `doi:10.1214/aos/1176350618`

Osborn, D., Chui, A., Smith, J., & Birchenhall, C. (1988). Seasonality and the order of integration for consumption. *Oxford Bulletin of Economics and Statistics*. 50(4), 361-377.

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
check_residuals(y,unit_root = "adf")
```

`cvm_bootstrap.test` *The Sieve Bootstrap Cramer Von Mises test for normality.*

Description

Performs the approximated Cramer Von Mises test of normality for univariate time series. Computes the p-value using Psaradakis and Vavra's (2020) sieve bootstrap procedure.

Usage

```
cvm_bootstrap.test(y, reps = 1000, h = 100, seed = NULL)
```

Arguments

y	a numeric vector or an object of the ts class containing a stationary time series.
reps	an integer with the total bootstrap repetitions.
h	an integer with the first burn-in sieve bootstrap replicates.
seed	An optional seed to use.

Details

Employs Cramer Von Mises test approximating the p-value using a sieve-bootstrap procedure, *Psaradakis, Z. and Vávra, M. (2020).*

Value

A list with class "h.test" containing the following components:

statistic:	the sieve bootstrap Cramer Von Mises' statistic.
p.value:	the p value for the test.
alternative:	a character string describing the alternative hypothesis.
method:	a character string "Sieve-Bootstrap Cramer Von Mises' test".
data.name:	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros.

References

Psaradakis, Z. and Vávra, M. (2020) Normality tests for dependent data: large-sample and bootstrap approaches. Communications in *Statistics-Simulation and Computation 49 (2)*. ISSN 0361-0918.

Bulmann, P. (1997). Sieve Bootstrap for time series. *Bernoulli*. 3(2), 123 -148.

Stephens, M.A. (1986): Tests based on EDF statistics. In: D'Agostino, R.B. and Stephens, M.A., eds.: Goodness-of-Fit Techniques. Marcel Dekker, New York.

See Also

vavra.test, sieve.bootstrap

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
cvm_bootstrap.test(y)
```

`elbouch.statistic` *Computes El Bouch, et al.'s z statistic.*

Description

Computes the El Bouch, Michel, & Comon's z test statistic for normality of a univariate or bivariate time series.

Usage

```
elbouch.statistic(y, x = NULL)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`x`	a numeric vector or an object of the `ts` class containing a stationary time series.

Details

This function computes Mardia's standardized 'z = (B - E_B)/ sd_B' statistic corrected by El Bouch, et al. (2022) for stationary bivariate time series. Where: 'B' is the square of a quadratic form of the process 'c(y, x)'; 'E_B' and 'sd_B' are the estimator's expected value and standard error respectively. If 'x' is set to 'NULL', the test computes the univariate counterpart.

Value

a real value with El Bouch test's statistic.

Author(s)

Asael Alonzo Matamoros.

References

El Bouch, S., Michel, O. & Comon, P. (2022). A normality test for Multivariate dependent samples. *Journal of Signal Processing*. Volume 201.

Mardia, K. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57 519-530

Lobato, I., & Velasco, C. (2004). A simple test of normality in time series. *Journal of econometric theory*. 20(4), 671-689.

See Also

`lobato.statistic`

Examples

```
# Genere an univariate stationary ARMA process
y = arima.sim(100,model = list(ar = 0.3))
elbouch.statistic(y)

# Generate a bivariate Gaussian random vector
x = rnorm(200)
y = rnorm(200)
elbouch.statistic(y = y, x = x)
```

`elbouch.test` *Computes El Bouch, et al.'s test for normality of multivariate dependent samples.*

Description

Computes the El Bouch, Michel, & Comon's test for normality of a bivariate dependent samples.

Usage

```
elbouch.test(y, x = NULL)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`x`	a numeric vector or an object of the `ts` class containing a stationary time series.

Details

This function computes El Bouch, et al. (2022) test for normality of bivariate dependent samples. If ‘x‘ is set to ‘NULL‘, the test computes the univariate counterpart. This test is a correction of Mardia's, (1970) multivariate skewness and kurtosis test for multivariate samples.

Value

A list with class `"h.test"` containing the following components:

`statistic:`	the El Bouch Z statistic.
`p.value:`	the p value for the test.
`alternative:`	a character string describing the alternative hypothesis.
`method:`	a character string “El Bouch, Michel & Comon's test”.
`data.name:`	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros.

References

El Bouch, S., Michel, O. & Comon, P. (2022). A normality test for Multivariate dependent samples. *Journal of Signal Processing*. Volume 201.

Mardia, K. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57 519-530

Lobato, I., & Velasco, C. (2004). A simple test of normality in time series. *Journal of econometric theory*. 20(4), 671-689.

See Also

lobato.test

Examples

```
# Generate an univariate stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
elbouch.test(y)

# Generate a bivariate Gaussian random vector
x = rnorm(200)
y = rnorm(200)
elbouch.test(y = y, x = x)
```

---

epps.statistic *Estimates the Epps statistic.*

Description

Estimates the Epps statistic minimizing the quadratic loss of the process' characteristic function in terms of the first two moments.

Usage

```
epps.statistic(y, lambda = c(1,2))
```

Arguments

y	a numeric vector or an object of the ts class containing a stationary time series.
lambda	a numeric vector for evaluating the characteristic function. This values could be selected by the user for a better test performance. By default, the values are ‘c(1,2)‘, another plausible option is to select random values.

Details

The Epps test minimize the process' empirical characteristic function using a quadratic loss in terms of the process two first moments. *Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014)* upgrade the test implementation by allowing the option of evaluating the characteristic function with random values.

This function is the equivalent of `Sub` in *Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014)*. This function uses a quadratic optimization solver implemented by *Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (2007)*.

Value

a real value with the Epps test's statistic.

Author(s)

Alicia Nieto-Reyes and Asael Alonzo Matamoros.

References

Epps, T.W. (1987). Testing that a stationary time series is Gaussian. *The Annals of Statistic*. 15(4), 1683-1698.

Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014). A random-projection based test of Gaussianity for stationary processes. *Computational Statistics & Data Analysis, Elsevier*, vol. 75(C), pages 124-141.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (2007). Numerical Recipes. The Art of Scientific Computing. *Cambridge University Press*.

See Also

`lobato.statistic`

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
epps.statistic(y)
```

`epps.test` *The asymptotic Epps and Pulley Test for normality.*

Description

Performs the asymptotic Epps test of normality for univariate time series. Computes the p-value using the asymptotic Gamma Distribution.

Usage

```
epps.test(y, lambda = c(1,2))
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`lambda`	a numeric vector for evaluating the characteristic function. This values could be selected by the user for a better test performance. By default, the values are 'c(1,2)', another plausible option is to select random values.

Details

The Epps test minimize the process' empirical characteristic function using a quadratic loss in terms of the process two first moments. *Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014)* upgrade the test implementation by allowing the option of evaluating the characteristic function with random values. The `amoebam()` function of *Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (2007)*, performs the optimal search.

Value

A list with class `"h.test"` containing the following components:

`statistic:`	the Epps statistic.
`parameter:`	the test degrees freedoms.
`p.value:`	the p value.
`alternative:`	a character string describing the alternative hypothesis.
`method:`	a character string "Epps test".
`data.name:`	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros and Alicia Nieto-Reyes.

References

Epps, T.W. (1987). Testing that a stationary time series is Gaussian. *The Annals of Statistic*. 15(4), 1683-1698.

Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014). A random-projection based test of Gaussianity for stationary processes. *Computational Statistics & Data Analysis, Elsevier*, vol. 75(C), pages 124-141.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (2007). Numerical Recipes. The Art of Scientific Computing. *Cambridge University Press*.

See Also

`lobato.test`

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
epps.test(y)

# Epps tests with random lambda values
y = arima.sim(100,model = list(ar = c(0.3,0.2)))
epps.test(y, lambda = rnorm(2,mean = 1,sd = 0.1))
```

---

epps_bootstrap.test *The Sieve Bootstrap Epps and Pulley test for normality.*

Description

Performs the approximated Epps and Pulley's test of normality for univariate time series. Computes the p-value using Psaradakis and Vavra's (2020) sieve bootstrap procedure.

Usage

```
epps_bootstrap.test(y, lambda = c(1,2), reps = 500, h = 100, seed = NULL)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`lambda`	a numeric vector for evaluating the characteristic function.
`reps`	an integer with the total bootstrap repetitions.
`h`	an integer with the first `burn-in` sieve bootstrap replicates.
`seed`	An optional `seed` to use.

Details

The Epps test minimize the process' empirical characteristic function using a quadratic loss in terms of the process two first moments, *Epps, T.W. (1987)*. Approximates the p-value using a sieve-bootstrap procedure *Psaradakis, Z. and Vávra, M. (2020)*.

Value

A list with class `"h.test"` containing the following components:

`statistic:`	the sieve bootstrap Epps and Pulley's statistic.
`p.value:`	the p value for the test.
`alternative:`	a character string describing the alternative hypothesis.
`method:`	a character string "Sieve-Bootstrap Epps' test".
`data.name:`	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros and Alicia Nieto-Reyes.

References

Psaradakis, Z. and Vávra, M. (2020) Normality tests for dependent data: large-sample and bootstrap approaches. Communications in *Statistics-Simulation and Computation 49 (2)*. ISSN 0361-0918.

Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014). A random-projection based test of Gaussianity for stationary processes. *Computational Statistics & Data Analysis, Elsevier*, vol. 75(C), pages 124-141.

Epps, T.W. (1987). Testing that a stationary time series is Gaussian. *The Annals of Statistic*. 15(4), 1683-1698.

See Also

`lobato.statistic`, `epps.test`

Examples

```
# Generating an stationary arma process
y = arima.sim(300, model = list(ar = 0.3))
epps_bootstrap.test(y, reps = 1000)
```

---

ggacf — `acf` *plot.*

Description

Plot of the auto-correlation function for a univariate time series.

Usage

```
ggacf(y, title = NULL)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`title`	a string with the plot's title.

Value

None.

Author(s)

Asael Alonzo Matamoros

Examples

```
x = rnorm(100)
ggacf(x)
```

---

gghist *Histogram with optional normal density functions.*

Description

Plots a histogram and density estimates using ggplot.

Usage

```
gghist(y, title = NULL, xlab = NULL, ylab = "counts", bins, add.normal = TRUE)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`title`	a string with the plot's title.
`xlab`	a string with the plot's x axis label. By default a NULL value.
`ylab`	a string with the plot's y axis label. By default a "counts" value.
`bins`	the number of bins to use for the histogram. Selected by default using the Friedman-Diaconis rule.
`add.normal`	a boolean value. Add a normal density function for comparison, by default `add.normal = TRUE`.

Value

None.

Author(s)

Rob J Hyndman

Examples

```
x = rnorm(100)
gghist(x,add.normal = TRUE)
```

---

ggnorm — `qqplot` *with normal* `qqline`

Description

Plot the quantile-quantile plot and quantile-quantile line using ggplot.

Usage

```
ggnorm(y, title = NULL, add.normal = TRUE)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`title`	a string with the plot's title.
`add.normal`	Add a normal density function for comparison.

Value

None.

Author(s)

Asael Alonzo Matamoros

Examples

```
x = rnorm(100)
ggnorm(x)
```

---

ggpacf — `pacf` *plot.*

Description

Plot of the partial autocorrelation function for a univariate time series.

Usage

```
ggpacf(y, title = NULL)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`title`	a string with the plot's title.

Value

None.

Author(s)

Mitchell O'Hara-Wild and Asael Alonzo Matamoros

Examples

```
x = rnorm(100)
ggpacf(x)
```

---

jb_bootstrap.test *The Sieve Bootstrap Jarque-Bera test for normality.*

Description

Performs the approximated Jarque Bera test of normality for univariate time series. Computes the p-value using Psaradakis and Vavra's (2020) sieve bootstrap procedure.

Usage

```
jb_bootstrap.test(y, reps = 1000, h = 100, seed = NULL)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`reps`	an integer with the total bootstrap repetitions.
`h`	an integer with the first `burn-in` sieve bootstrap replicates.
`seed`	An optional `seed` to use.

Details

Employs Jarque Bera skewness-kurtosis test approximating the p-value using a sieve-bootstrap procedure, *Psaradakis, Z. and Vávra, M. (2020)*.

Value

A list with class `"h.test"` containing the following components:

`statistic:`	the sieve bootstrap Jarque Bera's statistic.
`p.value:`	the p value for the test.
`alternative:`	a character string describing the alternative hypothesis.
`method:`	a character string "Sieve-Bootstrap Jarque Bera's test".
`data.name:`	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros.

References

Psaradakis, Z. and Vávra, M. (2020) Normality tests for dependent data: large-sample and bootstrap approaches. Communications in *Statistics-Simulation and Computation 49 (2)*. ISSN 0361-0918.

Bulmann, P. (1997). Sieve Bootstrap for time series. *Bernoulli*. 3(2), 123 -148.

J. B. Cromwell, W. C. Labys and M. Terraza (1994): Univariate Tests for Time Series Models, Sage, Thousand Oaks, CA, pages 20–22.

See Also

`vavra.test`, `sieve.bootstrap`

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
jb_bootstrap.test(y)
```

`Lm.test` *The Lagrange Multiplier test for arch effect.*

Description

Performs the Lagrange Multipliers test for homoscedasticity in a stationary process. The null hypothesis (H0), is that the process is homoscedastic.

Usage

```
Lm.test(y,lag.max = 2,alpha = 0.05)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`lag.max`	an integer with the number of used lags.
`alpha`	Level of the test, possible values range from 0.01 to 0.1. By default `alpha = 0.05` is used.

Details

The Lagrange Multiplier test proposed by *Engle (1982)* fits a linear regression model for the squared residuals and examines whether the fitted model is significant. So the null hypothesis is that the squared residuals are a sequence of white noise, namely, the residuals are homoscedastic.

Value

A list with class "h.test" containing the following components:

`statistic:`	the Lagrange multiplier statistic.
`parameter:`	the test degrees freedoms.
`p.value:`	the p value.
`alternative:`	a character string describing the alternative hypothesis.
`method:`	a character string “Lagrange Multiplier test”.
`data.name:`	a character string giving the name of the data.

Author(s)

A. Trapletti and Asael Alonzo Matamoros.

References

Engle, R. F. (1982). Auto-regressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*. 50(4), 987-1007.

McLeod, A. I. and W. K. Li. (1984). Diagnostic Checking ARMA Time Series Models Using Squared-Residual Auto-correlations. *Journal of Time Series Analysis.* 4, 269-273.

See Also

`arch.test`

Examples

```
# generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
Lm.test(y)
```

`lobato.statistic` *Computes the Lobato and Velasco statistic.*

Description

Computes the Lobato and Velasco’s statistic. This test proves a normality assumption in correlated data employing the skewness-kurtosis test statistic, but studentized by standard error estimates that are consistent under serial dependence of the observations.

Usage

```
lobato.statistic(y, c = 1)
```

Arguments

- `y` a numeric vector or an object of the `ts` class containing a stationary time series.
- `c` a positive real value that identifies the total amount of values used in the cumulative sum.

Details

This function is the equivalent of `GestadisticoVn` of *Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014).*

Value

A real value with the Lobato and Velasco test's statistic.

Author(s)

Alicia Nieto-Reyes and Asael Alonzo Matamoros.

References

Lobato, I., & Velasco, C. (2004). A simple test of normality in time series. *Journal of econometric theory*. 20(4), 671-689.

Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014). A random-projection based test of Gaussianity for stationary processes. *Computational Statistics & Data Analysis, Elsevier*, vol. 75(C), pages 124-141.

See Also

`epps.statistic`

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
lobato.statistic(y,3)
```

`lobato.test` *The asymptotic Lobato and Velasco's Test for normality.*

Description

Performs the asymptotic Lobato and Velasco's test of normality for univariate time series. Computes the p-value using the asymptotic Gamma Distribution.

Usage

```
lobato.test(y,c = 1)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`c`	a positive real value that identifies the total amount of values used in the cumulative sum.

Details

This test proves a normality assumption in correlated data employing the skewness-kurtosis test statistic, but studentized by standard error estimates that are consistent under serial dependence of the observations. The test was proposed by *Lobato, I., & Velasco, C. (2004)* and implemented by *Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014)*.

Value

A list with class `"h.test"` containing the following components:

`statistic:`	the Lobato and Velasco's statistic.
`parameter:`	the test degrees freedoms.
`p.value:`	the p-value for the test.
`alternative:`	a character string describing the alternative hypothesis.
`method:`	a character string “Lobato and Velasco's test”.
`data.name:`	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros and Alicia Nieto-Reyes.

References

Lobato, I., & Velasco, C. (2004). A simple test of normality in time series. *Journal of econometric theory*. 20(4), 671-689.

Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014). A random-projection based test of Gaussianity for stationary processes. *Computational Statistics & Data Analysis, Elsevier*, vol. 75(C), pages 124-141.

See Also

`lobato.statistic`,`epps.test`

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
lobato.test(y)
```

lobato_bootstrap.test *The Sieve Bootstrap Lobato and Velasco's Test for normality.*

Description

Performs the approximated Lobato and Velasco's test of normality for univariate time series. Computes the p-value using Psaradakis and Vavra's (2020) sieve bootstrap procedure.

Usage

```
lobato_bootstrap.test(y, c = 1, reps = 1000, h = 100, seed = NULL)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`c`	a positive real value that identifies the total amount of values used in the cumulative sum.
`reps`	an integer with the total bootstrap repetitions.
`h`	an integer with the first `burn-in` sieve bootstrap replicates.
`seed`	An optional `seed` to use.

Details

This test proves a normality assumption in correlated data employing the skewness-kurtosis test statistic proposed by *Lobato, I., & Velasco, C. (2004)*, approximating the p-value using a sieve-bootstrap procedure, *Psaradakis, Z. and Vávra, M. (2020)*.

Value

A list with class `"h.test"` containing the following components:

`statistic:`	the sieve bootstrap Lobato n Velasco's statistic.
`p.value:`	the p value for the test.
`alternative:`	a character string describing the alternative hypothesis.
`method:`	a character string "Sieve-Bootstrap Lobato's test".
`data.name:`	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros and Alicia Nieto-Reyes.

References

Psaradakis, Z. and Vávra, M. (2020) Normality tests for dependent data: large-sample and bootstrap approaches. Communications in *Statistics-Simulation and Computation 49 (2)*. ISSN 0361-0918.

Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014). A random-projection based test of Gaussianity for stationary processes. *Computational Statistics & Data Analysis, Elsevier*, vol. 75(C), pages 124-141.

Lobato, I., & Velasco, C. (2004). A simple test of normality in time series. *Journal of econometric theory*. 20(4), 671-689.

See Also

lobato.statistic,epps.test

Examples

```
# Generating an stationary arma process
y = arima.sim(1000,model = list(ar = 0.3))
lobato_bootstrap.test(y, reps = 1000)
```

---

normal.test *The normality test for stationary process*

Description

Perform a normality test. The null hypothesis (H0) is that the given data follows a stationary Gaussian process.

Usage

```
normal.test(y, normality = c("epps","lobato","vavra","rp","jb","ad","shapiro"),
                     alpha = 0.05)
```

Arguments

y	a numeric vector or an object of the `ts` class containing a stationary time series.
normality	A character string naming the desired test for checking normality. Valid values are `"epps"` for the Epps, `"lobato"` for Lobato and Velasco's,`"vavra"` for the Psaradakis and Vávra, `"rp"` for the random projections, `"jb"` for the Jarque and Bera, `"ad"` for Anderson Darling test, and `"shapiro"` for the Shapiro-Wilk's test. The default value is `"epps"` test.
alpha	Level of the test, possible values range from 0.01 to 0.1. By default `alpha = 0.05`

Details

"lobato", "epps", "vavras" and "rp" test are for testing normality in stationary process. "jb", "ad", and "shapiro" tests are for numeric data. In all cases, the alternative hypothesis is that y follows a Gaussian process. By default, alpha = 0.05 is used to select the more likely hypothesis.

Value

A list with class "h.test" containing the following components:

statistic:	the test statistic.
parameter:	the test degrees freedoms.
p.value:	the p-value for the test.
alternative:	a character string describing the alternative hypothesis.
method:	a character string with the test name.
data.name:	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros

References

Epps, T.W. (1987). Testing that a stationary time series is Gaussian. *The Annals of Statistic*. 15(4), 1683-1698.

Lobato, I., & Velasco, C. (2004). A simple test of normality in time series. *Journal of econometric theory*. 20(4), 671-689.

Psaradakis, Z. & Vávra, M. (2017). A distance test of normality for a wide class of stationary process. *Journal of Econometrics and Statistics*. 2, 50-60.

Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014). A random-projection based test of Gaussianity for stationary processes. *Computational Statistics & Data Analysis, Elsevier*, vol. 75(C), pages 124-141.

Patrick, R. (1982). An extension of Shapiro and Wilk's W test for normality to large samples. *Journal of Applied Statistics*. 31, 115-124.

Cromwell, J. B., Labys, W. C. & Terraza, M. (1994). Univariate Tests for Time Series Models. *Sage, Thousand Oaks, CA*. 20-22.

See Also

uroot.test, seasonal.test

Examples

```
#  stationary  ar process
y = arima.sim(100, model = list(ar = 0.3))
normal.test(y) # epps test

# normal random sample
```

```
y = rnorm(100)
normal.test(y, normality = "shapiro")

# exponential random sample
y = rexp(100)
normal.test(y, normality = "ad")
```

`random.projection` *Generate a random projection.*

Description

Generates a random projection of a univariate stationary stochastic process. Using a beta(shape1,shape2) distribution.

Usage

```
random.projection(y,shape1,shape2,seed = NULL)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`shape1`	an optional real value with the first shape parameters of the beta distribution.
`shape2`	an optional real value with the second shape parameters of the beta distribution.
`seed`	An optional `seed` to use.

Details

Generates one random projection of a stochastic process using a beta distribution. For more details, see: *Nieto-Reyes, A.,Cuesta-Albertos, J. & Gamboa, F. (2014).*

Value

a real vector with the projected stochastic process.

Author(s)

Alicia Nieto-Reyes and Asael Alonzo Matamoros.

References

Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014). A random-projection based test of Gaussianity for stationary processes. *Computational Statistics & Data Analysis, Elsevier*, vol. 75(C), pages 124-141.Result

Epps, T.W. (1987). Testing that a stationary time series is Gaussian. *The Annals of Statistic*. 15(4), 1683-1698.

Lobato, I., & Velasco, C. (2004). A simple test of normality in time series. *Journal of econometric theory*. 20(4), 671-689.

See Also

lobato.test epps.test

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
rp.test(y)
```

---

rp.sample *Generates a test statistics sample of random projections.*

Description

Generates a 2k sample of test statistics projecting the stationary process using the random projections procedure.

Usage

```
rp.sample(y, k = 1, pars1 = c(100,1), pars2 = c(2,7), seed = NULL)
```

Arguments

y	a numeric vector or an object of the `ts` class containing a stationary time series.
k	an integer that determines the '2k' random projections are used for every type of test. The 'pars1' argument generates the first 'k' projections, and 'pars2' generates the later 'k' projections. By default, `k = 1`.
pars1	an optional real vector with the shape parameters of the beta distribution used for the first 'k' random projections By default, `pars1 = c(100,1)` where, `shape1 = 100` and `shape2 = 1`.
pars2	an optional real vector with the shape parameters of the beta distribution used to compute the last 'k' random projections. By default, `pars2 = c(2,7)` where, `shape1 = 2` and `shape2 = 7`.
seed	An optional seed to use.

Details

The `rp.sample` function generates '2k' tests statistics by projecting the time series using '2k' stick breaking processes. First, the function samples 'k' stick breaking processes using `pars1` argument. Then, projects the time series using the sampled stick processes. Later, applies the Epps statistics to the odd projections and the Lobato and Velasco's statistics to the even ones. Analogously, the function performs the three steps using also `pars2` argument

The function uses beta distributions for generating the '2k' random projections. By default, uses a `beta(shape1 = 100,shape = 1)` distribution contained in `pars1` argument to generate the first 'k' projections. For the later 'k' projections the functions uses a `beta(shape1 = 2,shape = 7)` distribution contained in `pars2` argument.

The test was proposed by *Nieto-Reyes, A.,Cuesta-Albertos, J. & Gamboa, F. (2014).*

Value

A list with 2 real value vectors:

`lobato:` A vector with the Lobato and Velasco's statistics sample.

`epps:` A vector with the Epps statistics sample.

Author(s)

Alicia Nieto-Reyes and Asael Alonzo Matamoros

References

Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014). A random-projection based test of Gaussianity for stationary processes. *Computational Statistics & Data Analysis, Elsevier*, vol. 75(C), pages 124-141.

Epps, T.W. (1987). Testing that a stationary time series is Gaussian. *The Annals of Statistic*. 15(4), 1683-1698.

Lobato, I., & Velasco, C. (2004). A simple test of normality in time series. *Journal of econometric theory*. 20(4), 671-689.

See Also

`lobato.test`, `epps.test`

Examples

```
# Generating an stationary ARMA process
y = arima.sim(100,model = list(ar = 0.3))
rp.sample(y)
```

`rp.test` *The k random projections test for normality.*

Description

Performs the random projection test for normality. The null hypothesis (H0) is that the given data follows a stationary Gaussian process.

Usage

```
rp.test(y, k = 1, FDR = TRUE, pars1 = c(100,1), pars2  = c(2,7),
                seed = NULL)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`k`	an integer that determines the '2k' random projections are used for every type of test. The 'pars1' argument generates the first 'k' projections, and 'pars2' generates the later 'k' projections. By default, `k = 1`.
`FDR`	a logical value for mixing the p.values using a False discovery rate method. If `FDR = TRUE`, then the p.values are mixed using Benjamin and Yekutieli (2001) False discovery Rate method for dependent procedures, on the contrary, it applies Benjamini and Hochberg (1995) procedure. By default, `FDR = TRUE`.
`pars1`	an optional real vector with the shape parameters of the beta distribution used for the first 'k' random projections By default, `pars1 = c(100,1)` where, `shape1 = 100` and `shape2 = 1`.
`pars2`	an optional real vector with the shape parameters of the beta distribution used to compute the last 'k' random projections. By default, `pars2 = c(2,7)` where, `shape1 = 2` and `shape2 = 7`.
`seed`	An optional `seed` to use.

Details

The random projection test generates '2k' random projections of 'y'. Applies Epps statistics to the odd projections and Lobato and Velasco's statistics to the even ones. Computes the '2k' p.values using an asymptotic chi-square distribution with two degrees of freedom. Finally, mixes the p.values using a false discover rate procedure. By default, mixes the p.values using Benjamin and Yekutieli's (2001) method.

The function uses beta distributions for generating the '2k' random projections. By default, uses a `beta(shape1 = 100,shape = 1)` distribution contained in `pars1` argument to generate the first 'k' projections. For the later 'k' projections the functions uses a `beta(shape1 = 2,shape = 7)` distribution contained in `pars2` argument.

The test was proposed by *Nieto-Reyes, A.,Cuesta-Albertos, J. & Gamboa, F. (2014)*.

Value

A list with class `"h.test"` containing the following components:

`statistic:`	an integer value with the amount of projections per test.
`parameter:`	a text that specifies the p.value mixing FDR method.
`p.value:`	the FDR mixed p-value for the test.
`alternative:`	a character string describing the alternative hypothesis.
`method:`	a character string "k random projections test".
`data.name:`	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros and Alicia Nieto-Reyes.

References

Nieto-Reyes, A., Cuesta-Albertos, J. & Gamboa, F. (2014). A random-projection based test of Gaussianity for stationary processes. *Computational Statistics & Data Analysis, Elsevier*, vol. 75(C), pages 124-141.

Lobato, I., & Velasco, C. (2004). A simple test of normality in time series. *Journal of econometric theory*. 20(4), 671-689.

Benjamini, Y., and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*. 29, 1165–1188. Doi:10.1214/aos/1013699998.

Hochberg, Y. (1988). A sharper Bonferroni procedure for multiple tests of significance. *Biometrika*. 75, 800–803. Doi:10.2307/2336325.

Epps, T.W. (1987). Testing that a stationary time series is Gaussian. *The Annals of Statistic*. 15(4), 1683-1698.

See Also

`lobato.test`, `epps.test`

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
rp.test(y)
```

`seasonal.test` *The Seasonal unit root tests function*

Description

Perform a seasonal unit root test to check seasonality in a linear stochastic process

Usage

```
seasonal.test(y, seasonal = c("ocsb","ch","hegy"), alpha = 0.05)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`seasonal`	A character string naming the desired seasonal unit root test for checking seasonality. Valid values are `"ocsb"` for the Osborn, Chui, Smith, and Birchenhall, `"ch"` for the Canova and Hansen, and `"hegy"` for Hylleberg, Engle, Granger, and Yoo. The default value is `"ocsb"` for the Osborn, Chui, Smith, and Birchenhall test.
`alpha`	Level of the test, possible values range from 0.01 to 0.1. By default `alpha = 0.05` is used

Details

Several different tests are available: In the kpss test, the null hypothesis that y has a stationary root against a unit-root alternative. In the two remaining tests, the null hypothesis is that y has a unit root against a stationary root alternative. By default, alpha = 0.05 is used to select the more likely hypothesis.

Value

A list with class "h.test" containing the following components:

statistic:	the test statistic.
parameter:	the test degrees freedoms.
p.value:	the p-value for the test.
alternative:	a character string describing the alternative hypothesis.
method:	a character string with the test name.
data.name:	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros

References

Osborn, D., Chui, A., Smith, J., & Birchenhall, C. (1988). Seasonality and the order of integration for consumption. *Oxford Bulletin of Economics and Statistics*. 50(4), 361-377.

Canova, F. & Hansen, B. (1995). Are Seasonal Patterns Constant over Time? A Test for Seasonal Stability. *Journal of Business and Economic Statistics*. 13(3), 237-252.

Hylleberg, S., Engle, R., Granger, C. & Yoo, B. (1990). Seasonal integration and cointegration. *Journal of Econometrics* 44(1), 215-238.

See Also

normal.test, uroot.test

Examples

```
#  stationary  ar process
y = ts(rnorm(100),frequency = 6)
seasonal.test(y)
```

`shapiro_bootstrap.test` *The Sieve Bootstrap Shapiro test for normality.*

Description

Performs the approximated Shapiro test for normality for univariate time series. Computes the p-value using Psaradakis and Vavra's (2020) sieve bootstrap procedure.

Usage

```
shapiro_bootstrap.test(y, reps = 1000, h = 100, seed = NULL)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`reps`	an integer with the total bootstrap repetitions.
`h`	an integer with the first `burn-in` sieve bootstrap replicates.
`seed`	An optional `seed` to use.

Details

Employs the Shapiro test approximating the p-value using a sieve-bootstrap procedure, *Psaradakis, Z. and Vávra, M. (2020)*.

Value

A list with class `"h.test"` containing the following components:

`statistic:`	the sieve bootstrap Shapiro's statistic.
`p.value:`	the p value for the test.
`alternative:`	a character string describing the alternative hypothesis.
`method:`	a character string "Sieve-Bootstrap Shapiro's test".
`data.name:`	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros.

References

Psaradakis, Z. and Vávra, M. (2020) Normality tests for dependent data: large-sample and bootstrap approaches. Communications in *Statistics-Simulation and Computation 49 (2)*. ISSN 0361-0918.

Bulmann, P. (1997). Sieve Bootstrap for time series. *Bernoulli*. 3(2), 123 -148.

Patrick Royston (1982). An extension of Shapiro and Wilk's W test for normality to large samples. *Applied Statistics*, 31, 115–124. Doi:10.2307/2347973.

See Also

vavra.test, sieve.bootstrap

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
jb_bootstrap.test(y)
```

---

sieve.bootstrap *Generates a sieve bootstrap sample.*

Description

The function generates a sieve bootstrap sample for a univariate linear stochastic process.

Usage

```
sieve.bootstrap(y,reps = 1000,pmax = NULL,h = 100,seed = NULL)
```

Arguments

y	a numeric vector or an object of the `ts` class containing a stationary time series.
reps	an integer with the total bootstrap repetitions.
pmax	an integer with the max considered lags for the generated `ar(p)` process. By default, `pmax = NULL`.
h	an integer with the first `burn-in` sieve bootstrap replicates.
seed	An optional seed to use.

Details

simulates bootstrap samples for the stochastic process `y`, using a stationary auto-regressive model of order `"pmax"`, `AR(pmax)`. If `pmax = NULL` (*default*), the function estimates the process maximum lags using an `AIC` as a model selection criteria.

Value

A matrix or `reps` row and `n` columns, with the sieve bootstrap sample and `n` the time series length.

Author(s)

Asael Alonzo Matamoros.

References

Bulmann, P. (1997). Sieve Bootstrap for time series. *Bernoulli*. 3(2), 123 -148.

See Also

lobato.test, epps.test.

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
M = sieve.bootstrap(y)
```

---

uroot.test — *The Unit root tests function.*

Description

Perform a unit root test to check stationary in a linear stochastic process.

Usage

```
uroot.test(y, unit_root = c("adf","kpss","pp","box"), alpha = 0.05)
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`unit_root`	A character string naming the desired unit root test for checking stationary. Valid values are `"adf"` for the Augmented Dickey-Fuller, `"pp"` for the Phillips-Perron, `"kpss"` for Kwiatkowski, Phillips, Schmidt, and Shin, and `"box"` for the Ljung-Box. The default value is `"adf"` for the Augmented Dickey-Fuller test.
`alpha`	Level of the test, possible values range from 0.01 to 0.1. By default `alpha = 0.05` is used.

Details

Several different tests are available: In the `kpss` test, the null hypothesis that `y` has a stationary root against a unit-root alternative. In the two remaining tests, the null hypothesis is that `y` has a unit root against a stationary root alternative. By default, `alpha = 0.05` is used to select the more likely hypothesis.

Value

A list with class `"h.test"` containing the following components:

`statistic:`	the test statistic.
`parameter:`	the test degrees freedoms.
`p.value:`	the p-value for the test.
`alternative:`	a character string describing the alternative hypothesis.
`method:`	a character string with the test name.
`data.name:`	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros and A. Trapletti.

References

Dickey, D. & Fuller, W. (1979). Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *Journal of the American Statistical Association*. 74, 427-431.

Kwiatkowski, D., Phillips, P., Schmidt, P. & Shin, Y. (1992). Testing the Null Hypothesis of Stationarity against the Alternative of a Unit Root, *Journal of Econometrics*. 54, 159-178.

Phillips, P. & Perron, P. (1988). Testing for a unit root in time series regression, *Biometrika*. 72(2), 335-346.

Ljung, G. M. & Box, G. E. P. (1978). On a measure of lack of fit in time series models. *Biometrika*. 65, 297-303.

See Also

`normal.test`, `seasonal.test`

Examples

```
#  stationary  ar process
y = arima.sim(100,model = list(ar = 0.3))
uroot.test(y)

# a random walk process
y = cumsum(y)
uroot.test(y, unit_root = "pp")
```

`vavra.sample` *Vávra test's sieve Bootstrap sample for Anderson Darling statistic*

Description

Generates a sieve bootstrap sample of the Anderson-Darling statistic test.

Usage

```
vavra.sample(y, normality = c("ad","lobato","jb","cvm","shapiro","epps"),
                      reps = 1000, h = 100, seed = NULL, c = 1, lambda = c(1,2))
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`normality`	A character string naming the desired test for checking normality. Valid values are `"epps"` for the Epps, `"lobato"` for Lobato and Velasco's, `"jb"` for the Jarque and Bera, `"ad"` for Anderson Darling test,`"cvm"` for the Cramer Von Mises' test, and `"shapiro"` for the Shapiro's test. The default value is `"ad"` test.
`reps`	an integer with the total bootstrap repetitions.
`h`	an integer with the first `burn-in` sieve bootstrap replicates.
`seed`	An optional `seed` to use.
`c`	a positive real value used as argument for the Lobato's test.
`lambda`	a numeric vector used as argument for the Epps's test.

Details

The Vávra test approximates the empirical distribution function of the Anderson-Darlings statistic, using a sieve bootstrap approximation. The test was proposed by *Psaradakis, Z. & Vávra, M (20.17)*.

This function is the equivalent of `xarsieve` of *Psaradakis, Z. & Vávra, M (20.17)*.

Value

A numeric array with the Anderson Darling sieve bootstrap sample

Author(s)

Asael Alonzo Matamoros.

References

Psaradakis, Z. and Vávra, M. (2020) Normality tests for dependent data: large-sample and bootstrap approaches. Communications in *Statistics-Simulation and Computation 49 (2)*. ISSN 0361-0918.

Psaradakis, Z. & Vávra, M. (2017). A distance test of normality for a wide class of stationary process. *Journal of Econometrics and Statistics*. 2, 50-60.

Bulmann, P. (1997). Sieve Bootstrap for time series. *Bernoulli*. 3(2), 123 -148.

See Also

`epps.statistic`, `lobato.statistic`

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
adbs = vavra.sample(y)
mean(adbs)
```

`vavra.test` *The Psaradakis and Vávra test for normality.*

Description

Performs the Psaradakis and Vávra distance test for normality. The null hypothesis (H0), is that the given data follows a Gaussian process.

Usage

```
vavra.test(y, normality = c("ad","lobato","jb","cvm","epps"),
                  reps = 1000, h = 100, seed = NULL, c = 1, lambda = c(1,2))
```

Arguments

`y`	a numeric vector or an object of the `ts` class containing a stationary time series.
`normality`	A character string naming the desired test for checking normality. Valid values are `"epps"` for the Epps, `"lobato"` for Lobato and Velasco's, `"jb"` for the Jarque and Bera, `"ad"` for Anderson Darling test, and `"cvm"` for the Cramer Von Mises' test. The default value is `"ad"` test.
`reps`	an integer with the total bootstrap repetitions.
`h`	an integer with the first `burn-in` sieve bootstrap replicates.
`seed`	An optional `seed` to use.
`c`	a positive real value used as argument for the Lobato's test.
`lambda`	a numeric vector used as argument for the Epps's test.

Details

The Psaradakis and Vávra test approximates the empirical distribution function of the Anderson Darling's statistic, using a sieve bootstrap approximation. The test was proposed by *Psaradakis, Z. & Vávra, M. (20.17).*

Value

A list with class `"h.test"` containing the following components:

`statistic:`	the sieve bootstrap A statistic.
`p.value:`	the p value for the test.
`alternative:`	a character string describing the alternative hypothesis.
`method:`	a character string "Psaradakis and Vávra test".
`data.name:`	a character string giving the name of the data.

Author(s)

Asael Alonzo Matamoros.

References

Psaradakis, Z. and Vávra, M. (2020) Normality tests for dependent data: large-sample and bootstrap approaches. Communications in *Statistics-Simulation and Computation 49 (2)*. ISSN 0361-0918.

Psaradakis, Z. & Vávra, M. (2017). A distance test of normality for a wide class of stationary process. *Journal of Econometrics and Statistics*. 2, 50-60.

Bulmann, P. (1997). Sieve Bootstrap for time series. *Bernoulli*. 3(2), 123 -148.

See Also

`lobato.test`, `epps.test`

Examples

```
# Generating an stationary arma process
y = arima.sim(100,model = list(ar = 0.3))
vavra.test(y)
```

Index