Package 'ts.extend'

November 14, 2020

Type Package

Title Stationary Gaussian ARMA Processes and Other Time-Series Utilities

Version 0.1.1

Date 2020-11-14

Author Ben O'Neill [aut, cre]

Maintainer Ben O'Neill <ben.oneill@hotmail.com>

Description Stationary Gaussian ARMA processes and the stationary 'GARMA' distribution are fundamental in time series analysis. Here we give utilities to compute the auto-covariance/auto-correlation for a stationary Gaussian ARMA process, as well as the probability functions (density, cumulative distribution, random generation) for random vectors from this distribution. We also give functions for the spectral intensity, and the permutation-spectrum test for testing a time-series vector for the presence of a signal.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports graphics, grDevices

Suggests ggplot2, gridExtra, mvtnorm

URL https://github.com/ben-oneill/ts.extend

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-11-14 21:50:03 UTC

R topics documented:

`ARMA.autocov` *Auto-covariance/auto-correlation function for the stationary ARMA model*

Description

This function computes a vector of output values from the auto-covariance/auto-correlation function for a stationary auto-regressive moving-average (ARMA) model. The user specifies the vector size `n` and the function returns a vector of auto-covariance/ auto-correlation values at all lags `Lag[0]`, . . . ,`Lag[n-1]`. The function requires the model to be stationary, which means that the vector of auto-regression coefficients must give an auto-regressive characteristic polynomial with roots outside the unit circle.

Usage

```
ARMA.autocov(n, ar = numeric(0), ma = numeric(0), corr = FALSE)
```

Arguments

`n`	Positive integer giving the number of consecutive values in the time-series (output is a vector of length `n`)
`ar`	Vector of auto-regressive coefficients (all roots of AR characteristic polynomial must be outside the unit circle)
`ma`	Vector of moving-average coefficients
`corr`	Logical; if `TRUE` the function returns the auto-correlation function; if `FALSE` the function returns the auto-covariance function

Examples

```
data(garma)

AR <- c(0.8, -0.2)
MA <- c(0.6,  0.3)
#Compute the auto-correlation function
ARMA.autocov(n = 6, ar = AR, ma = MA, corr = TRUE)
```

ARMA.var *Covariance/correlation matrix for the stationary ARMA model*

Description

This function computes the covariance/correlation matrix for a stationary auto-regressive moving-average (ARMA) model. The user specifies the matrix size n and the function returns a matrix of covariance/correlation values at all times Time[1],... ,Time[n] (in the case where conditioning values are specified using the condvals argument, only the time values for non-conditional values are included). The function requires the model to be stationary, which means that the vector of auto-regression coefficients must give an auto-regressive characteristic polynomial with roots outside the unit circle.

Usage

```
ARMA.var(
  n,
  condvals = as.numeric(NA),
  ar = numeric(0),
  ma = numeric(0),
  corr = FALSE
)
```

Arguments

n	Positive integer giving the number of values in the time-series (output variance matrix is an n x n matrix)
condvals	Either a single value NA or a numeric vector with n elements; numeric entries are conditioning values for the generated vector
ar	Vector of auto-regressive coefficients (all roots of AR characteristic polynomial must be outside the unit circle)
ma	Vector of moving-average coefficients
corr	Logical; if TRUE the function returns the correlation matrix; if FALSE the function returns the covariance matrix

dGARMA *Density function for the stationary GARMA distribution*

Description

This function computes the probability density function from the stationary Gaussian auto-regressive moving-average (GARMA) distribution. The user specifies a vector x giving a single time-series vector, or a matrix x giving one time-series vector in each row, and the function returns the vector of cumulative probabilities corresponding to the input time-series vectors. By default the function generates from the marginal GARMA distribution, but the user may give conditioning indicators in the cond vector to compute the conditional density where some of the elements in the input vectors are conditioning values.

Usage

```
dGARMA(
  x,
  cond = FALSE,
  mean = 0,
  errorvar = 1,
  ar = numeric(0),
  ma = numeric(0),
  log = FALSE
)
```

Arguments

x	A vector or matrix of time-series values (if a matrix, each time-series should be one row of the matrix)
cond	Either a single logical value FALSE or a logical vector with the same number of elements; as each time-series vector; each logical value indicates whether the density is conditional on the associated time-series value in x.
mean	The mean parameter
errorvar	The error variance parameter
ar	Vector of auto-regressive coefficients (all roots of AR characteristic polynomial must be outside the unit circle)
ma	Vector of moving-average coefficients
log	Logical; if TRUE the function returns the log-density; if FALSE the function returns the density

Examples

```
data(garma)
AR <- c(0.8, -0.2)
MA <- c(0.6,  0.3)

#Compute the density of the GARMA output
(DENSITY <- dGARMA(SERIES, ar = AR, ma = MA))
```

garma *Simulated example data set*

Description

A simulated set of 16 time series over 30 time points.

Details

Created using `set.seed(1014745590)`

intensity *Compute the spectral intensity of a time-series vector/matrix*

Description

This function computes the spectral intensity of a time series vector/matrix. The user inputs the time-series vector or matrix and specifies whether it is to be centered and/or scaled. Centering subtracts the sample mean of the vector prior to conversion into frequency space; this sets the intensity of the signal to zero at the zero frequency. Scaling scales the intensity so the that norm of the intensity vector is equal to the number of values in the time-series.

Usage

```
intensity(
  x,
  centered = TRUE,
  centred = centered,
  scaled = TRUE,
  nyquist = TRUE
)
```

Arguments

`x`	A vector of time-series values
`centered`	Logical; if `TRUE` the time-series vector is centred (mean removed) before computing its intensity
`centred`	An alternative name for the `centered` input
`scaled`	Logical; if `TRUE` the intensity measure is scaled so that its norm is equal to the number of values in the time-series
`nyquist`	Logical; if `TRUE` the intensity vector is reduces for real time-series vectors to limit it to frequencies in the Nyquist range

Examples

```
data(garma)
intensity(SERIES1)
```

---

pGARMA *Cumulative distribution function for the stationary GARMA distribution*

Description

This function computes the cumulative distribution function from the stationary Gaussian auto-regressive moving-average (GARMA) distribution. The user specifies a vector x giving a single time-series vector, or a matrix x giving one time-series vector in each row, and the function returns the vector of cumulative probabilities corresponding to the input time-series vectors. By default the function generates from the marginal GARMA distribution, but the user may give conditioning indicators in the cond vector to compute the conditional density where some of the elements in the input vectors are conditioning values.

Usage

```
pGARMA(
  x,
  cond = FALSE,
  mean = 0,
  errorvar = 1,
  ar = numeric(0),
  ma = numeric(0),
  log = FALSE
)
```

Arguments

x	A vector or matrix of time-series values (if a matrix, each time-series should be one row of the matrix)
cond	Either a single logical value FALSE or a logical vector with the same number of elements; as each time-series vector; each logical value indicates whether the density is conditional on the associated time-series value in x.
mean	The mean parameter
errorvar	The error variance parameter
ar	Vector of auto-regressive coefficients (all roots of AR characteristic polynomial must be outside the unit circle)
ma	Vector of moving-average coefficients

log	Logical; if TRUE the function returns the log-probability; if FALSE the function returns the probability
	data(garma) AR <- c(0.8, -0.2) MA <- c(0.6, 0.3)
	#Compute the cumulative probability of the GARMA output (PROBS <- pGARMA(SERIES, ar = AR, ma = MA))

`plot.intensity` *Plot scatterplot matrix of intensity vectors*

Description

This is a custom plot function that operates on objects of class `intensity` (which is the output generated from the `intensity` function). This plot function generates a scatterplot matrix of intensity vectors using either `ggplot` or `base` graphics. The user must input either a single intensity vector `x` or a matrix `x` where each row is one intensity vector. The function generates a scatterplot matrix showing each of the intensity barplots. The user may choose to print or assign the object, or both. Since the function generates a scatterplot of intensity plots, there are certain limits in the output. If the user attempts to generate the plot for a time-series matrix with more than 36 intensity vectors, the user will be prompted to continue. The prompts can be removed in the arguments of the function.

Usage

```
## S3 method for class 'intensity'
plot(x, ggplot = TRUE, print = TRUE, user.prompt = TRUE, ...)
```

Arguments

x	A vector or matrix of intensity values (if a matrix, each intensity vector should be one row of the matrix)
ggplot	Logical; if TRUE the scatterplot is a `ggplot` object; if FALSE it is a `base` plot object
print	Logical; if TRUE the scatterplot is printed
user.prompt	Logical; if TRUE the user will be prompted for choices when the number if intensity vectors is large
...	unused

Examples

```
data(garma)
INT     <- intensity(SERIES1)
plot(INT)
```

plot.spectrum.test *Plot of the Permutation-Spectrum Test*

Description

This function generates a dual plot showing the results of the permutation-spectrum testusing either ggplot or base graphics. The user must input a test object produced by the spectrum.test function. The function produces a dual plot showing the scaled intensity of the time-series vector and the simulated null distribution of the maximum scaled intensity under the null hypothesis of an IID vector. The plots also report the value of the maximum scaled intensity and the resulting p-value for the test. This dual plot forms a useful companion to the permutation-spectrum test; it allows the user to visualise the simulated null distribution and test statistic.

Usage

```
## S3 method for class 'spectrum.test'
plot(x, ggplot = TRUE, print = TRUE, ...)
```

Arguments

x	A spectrum.test object produced by the spectrum.test function
ggplot	Logical; if TRUE the scatterplot is a ggplot object; if FALSE it is a base plot object
print	Logical; if TRUE the scatterplot is printed
...	unused

Examples

```
data(garma)

#Show the intensity of a time-series vector
TEST <- spectrum.test(SERIES1, sims = 100)
plot(TEST)
```

plot.time.series *Plot scatterplot matrix of time-series vectors*

Description

This is a custom plot function that operates on objects of class `time.series` (which is the output generated from the `rGARMA` function). This plot function generates a scatterplot matrix of time-series vectors using either `ggplot` or `base` graphics. The user must input either a single time-series vector `x` or a matrix `x` where each row is one time-series vector. The function generates a scatterplot matrix showing each of the time-series. If the plot is generated using `ggplot` graphics then the user has an option to include a background showing all points from all the series in grey. The user may choose to print or assign the object, or both. Since the function generates a scatterplot of time-series plots, there are certain limits in the output. If the user attempts to generate the plot for a time-series matrix with more than 36 time-series, the user will be prompted to continue and prompted about whether they want to include the background. The prompts can be removed in the arguments of the function.

Usage

```
## S3 method for class 'time.series'
plot(
  x,
  ggplot = TRUE,
  background = TRUE,
  print = TRUE,
  user.prompt = TRUE,
  ...
)
```

Arguments

`x`	A vector or matrix of time-series values (if a matrix, each time-series should be one row of the matrix)
`ggplot`	Logical; if `TRUE` the scatterplot is a `ggplot` object; if `FALSE` it is a `base` plot object
`background`	Logical; if `TRUE` then each `ggplot` scatterplot will include background points for all the time-series vectors
`print`	Logical; if `TRUE` the scatterplot is printed
`user.prompt`	Logical; if `TRUE` the user will be prompted for choices when the number if time-series is large
`...`	unused
	#@examples
	data(garma) plot(SERIES)

rGARMA *Generate random vectors from the stationary GARMA distribution*

Description

This function generates random vectors from the stationary Gaussian auto-regressive moving-average (GARMA) distribution. The user specifies the number of vectors n and their dimension m and the function returns an n x m matrix of generated time-series from the GARMA distribution with the specified parameters. By default the function generates from the marginal GARMA distribution, but the user may give conditional values in the condvals vector to generate from the associated conditional distribution (non-conditional values in this vector are given as NA).

Usage

```
rGARMA(
  n,
  m,
  condvals = as.numeric(NA),
  mean = 0,
  errorvar = 1,
  ar = numeric(0),
  ma = numeric(0)
)
```

Arguments

n	Positive integer giving the number of random vectors to generate
m	Positive integer giving the dimension of the random vectors to generate (i.e., the number of values in each time-series)
condvals	Either a single value NA or a numeric vector with m elements; numeric entries are conditioning values for the generated vector
mean	The mean parameter
errorvar	The error variance parameter
ar	Vector of auto-regressive coefficients (all roots of AR characteristic polynomial must be outside the unit circle)
ma	Vector of moving-average coefficients

Examples

```
#Set the model parameters
AR <- c(0.8, -0.2)
MA <- c(0.6,  0.3)
#Generate random time-series from the GARMA distribution
SERIES <- rGARMA(n = 16, m = 30, ar = AR, ma = MA)

#Set the conditional values
```

```
CONDVALS     <- rep(NA, 30)
CONDVALS[1]  <- -4
CONDVALS[12] <-  0
CONDVALS[30] <-  4

#Generate and plot random time-series from the GARMA distribution
SERIES.COND <- rGARMA(n = 16, m = 30, ar = AR, ma = MA, condvals = CONDVALS)
```

`spectrum.test` *Permutation-spectrum test for time-series data*

Description

Computes the permutation-spectrum test to detect a periodic signal in a real or complex time-series vector. The null hypothesis for the test is that the values in the time-series vector are independent and identically distributed with no signal, and the alternative hypothesis is that the time-series has at least one remaining periodic signal. The test statistic is the maximum scaled intensity of the observed time-series vector. The p-value for the test is the probability of observing a maximum scaled intensity at least as large as the observed value under the null distribution of exchangeability. The null distribution for the test is simulated by applying random permutations to the observed time-series. The number of simulations is controlled by the `sims` parameter.

Usage

```
spectrum.test(x = NULL, sims = 10^6, progress = TRUE)
```

Arguments

`x`	A vector of time-series values (must have at least two data points)
`sims`	Positive integer for the number of simulations to perform in the test
`progress`	Logical; if `TRUE` the function uses a progress bar to track its simulations

Examples

```
data(garma)

#Show the intensity of a time-series vector
spectrum.test(SERIES1, sims = 100)
```

Index