Package: pRSR (via r-universe)

October 31, 2024

Type Package

Title Test of Periodicity using Response Surface Regression

Version 3.1.1

Date 2016-05-13

Author M. S. Islam

Maintainer M. S. Islam <shahed-sta@sust.edu>

Depends R (>= 2.10)

Description Tests periodicity in short time series using response surface regression.

Classification/ACM G.3, G.4, I.5.1

Classification/MSC 62M10, 91B84

LazyData yes

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-05-16 19:31:44

Contents

pRSR-package *It tests periodicity for any series using response surface regression (RSR).*

Description

It tests periodicity for any series using response surface regression (RSR). Whole response surface is integrated in the package. Therefore, one can easily get the pvalue for testing periodicity for any series length. However, the main focus of this algorithm is for short series. Plot for response surface regression for different quantile values can also be obtained.

Details

Package: pRSR Type: Package Title: Test of Periodicity using Response Surface Regression Version: 3.1.1 Date: 2016-05-12 A

Author(s)

M. S. Islam Maintainer: M. S. Islam <shahed-sta@sust.edu>

References

Islam, M.S. (2008). Peridocity, Change Detection and Prediction in Microarrays. Ph.D. Thesis, The University of Western Ontario.

MacKinnon, J. G. (2002). Computing numerical distribution functions in econometrics. In proceedings of High Performance Computing Systems and Applications, edited by Pollard, A., Mewhort, D. J. and Weaver, D. F. Springer US. Vol. 451, 455-471.

Examples

```
# #Testing periodicity
z<-SimulateHReg(20, f=2.5/20, 1, 2)
pvalrsr(z)  # finding p-value using RSR

# For comparing with Fisher's g test
# library(GeneCycle)
# fisher.g.test(z) # Fisher's g test

# Plot for 75%, 90% and 95% quantiles.
plotrsr(n=10:50, q=c(75,90,95))
```

`FitHReg` *Fits Three Parameter Harmonic Regression*

Description

Estimates A, B and f in the harmonic regression, y(t)=mu+A*cos(2*pi*f*t)+B*sin(2*pi*f*t)+e(t) using LS.

Usage

```
FitHReg(y, t = 1:length(y), nf=150)
```

Arguments

`y`	series
`t`	time points
`nf`	nf, number of frequencies to enumerate

Details

Program is interfaced to C for efficient computation.

Value

Object of class "HReg" produced. This is a list with components: 'coefficients', 'residuals', 'Rsq', 'fstatistic', 'sigma', 'freq', 'LRStat' corresponding to the 3 regression coefficients, residuals, R-squared, F-statistic, residual sd, optimal frequency and LR-test statistic for null hypothesis white noise.

References

Islam, M.S. (2008). Peridocity, Change Detection and Prediction in Microarrays. Ph.D. Thesis, The University of Western Ontario.

See Also

`GetFitHReg`

Examples

```
z<-SimulateHReg(10, f=2.5/10, 1, 2)
FitHReg(z)
```

GetFitHReg *Compute loglikelihood ratio test statistic*

Description

The loglikelihood ratio test statistic is computed for testing for periodicity

Usage

```
GetFitHReg(y, t, nf=150)
```

Arguments

y	vector containing the series
t	vector of corresponding time points
nf	nf, number of frequencies to enumerate

Details

This function interfaces with C code for fast evaluation.

Value

LR statistic and estimated frequency

References

Islam, M.S. (2008). Peridocity, Change Detection and Prediction in Microarrays. Ph.D. Thesis, The University of Western Ontario.

Examples

```
#Simple Examples
z<-SimulateHReg(10, f=2.5/10, 1, 2)
GetFitHReg(z)
t<-seq(2,20,2)
GetFitHReg(y=z, t=t)
GetFitHReg(z, nf=25)
```

plotrsr *Plot of response surface regression for different quantile*

Description

It produces a single plot of response surface regression for different quantile values.

Usage

```
plotrsr(n = 20:50, q = c(90, 95),...)
```

Arguments

n	Sequence of series size for the plot
q	Vector or single quantile value
...	Further arguments for function lines()

Value

Plot of RSR

Author(s)

M. S. Islam

References

MacKinnon, J. G. (2002). Computing numerical distribution functions in econometrics. In proceedings of High Performance Computing Systems and Applications, edited by Pollard, A., Mewhort, D. J. and Weaver, D. F. Springer US. Vol. 451, 455-471.

See Also

pvalrsr

Examples

```
# Plot for 75%, 90% and 95% quantiles.
plotrsr(n=10:50, q=c(75,90,95))
# Plot for 80%, 90%, 95% and 99% quantiles.
# We use color red and dashed line
plotrsr(n=10:50, q=c(80,90,95, 99), col=2, lty=3)
```

pvalrsr *Finds pvalue for testing periodicity using RSR.*

Description

It finds pvalue for tesing periodicity for any series using RSR.

Usage

```
pvalrsr(x,  t=1:length(x), nf=150, Numpq = 11)
```

Arguments

x	series to be tested for periodicity
t	vector of corresponding time points
nf	number of frequencies to enumerate
Numpq	Numebr of indices for interpolation in RSR

Details

A full RSR is integral part of the package. This was done using likelihood ratio statistic for simulated series from white noise process. For more information about the procedure, please see the first reference.

Value

pvalue

Author(s)

M. S. Islam

References

Islam, M.S. (2008). Peridocity, Change Detection and Prediction in Microarrays. Ph.D. Thesis, The University of Western Ontario.

MacKinnon, J. G. (2001). Computing numerical distribution functions in econometrics. In proceedings of High Performance Computing Systems and Applications, edited by Pollard, A., Mewhort, D. J. and Weaver, D. F. Springer US. Vol. 451, 455-471.

Examples

```
# Non-Fourier frequency
z<-SimulateHReg(20, f=2.5/20, 1, 2)
pvalrsr(z)  # finding p-value using RSR

# For comparing with Fisher's g test
# library(GeneCycle)
# fisher.g.test(z) # Fisher's g test


# Fourier frequency
y<-SimulateHReg(20, f=2/20, 1, 2)
pvalrsr(y)  # finding p-value using RSR
# For comparing with Fisher's g test
# library(GeneCycle)
# fisher.g.test(z) # Fisher's g test
```

`SimulateAR1` *Simulate AR(1) series*

Description

An AR(1) series with mean zero and variance 1 and with autocorrelation paramater phi is simulated.

Usage

```
SimulateAR1(n, phi)
```

Arguments

`n`	length of series
`phi`	autocorrelation parameter

Details

The model equation is: z[t] = phi*z[t-1]+a[t], where z[1] is N(0,1) and a[t] are NID(0, siga), $siga = \sqrt{(1/(1-phi^2))}$.

Value

autocorrelated time series of length n

See Also

`FitHReg`, `SimulateHReg`

Examples

```
e<-SimulateAR1(10^4, phi=0.8)
mean(e)
sd(e)
acf(e, lag.max=5, plot=FALSE)
```

SimulateHReg *Simulate Harmonic Regression*

Description

Simulates a harmonic regression. Possible error distributions are normal, t(5), t(5,6), AR1.

Usage

```
SimulateHReg(n, f, A, B, simPlot = FALSE, Dist = "n", phi = 0.3)
```

Arguments

n	length of series
f	frequency
A	cosine amplitude
B	sine amplitude
simPlot	plot simulated series
Dist	one of "n" for normal, "t" for t(5), "s" skewed t(5,6), "a" autocorrelated AR1 with parameter phi
phi	only used if AR1 error distribution is selected

Details

This will simulate harmonic hegression

Value

vector of length n, simulated harmonic series

See Also

FitHReg

Examples

```
z<-SimulateHReg(10, f=2.5/10, 1, 2)
FitHReg(z)
```

TPout *Utility function*

Description

Utility function that contains all objects for RSR

Usage

```
data(TPout)
```

Format

The format is: List of 4 $ qhatM : num [1:341, 1:195] 3.6 3.68 3.73 3.78 3.81 ... $ probs : num [1:341] 1e-04 2e-04 3e-04 4e-04 5e-04 6e-04 7e-04 8e-04 9e-04 1e-03 ... $ n : num [1:33] 6 8 10 12 14 16 18 20 22 24 ... $ MeanVar:List of 1 ..$:'data.frame': 33 obs. of 682 variables: .. ..$ Qt0.01 : num [1:33] 3.46 3.49 3.53 3.63 3.72 ... .$ Var0.01 : num [1:33] 0.00866 0.01268 0.01014 0.01387 0.01532 ... .$ Qt0.02 : num [1:33] 3.52 3.55 3.6 3.7 3.8 ... .$ Var0.02 : num [1:33] 0.00753 0.00985 0.0067 0.01003 0.01245 ... .$ Qt0.03 : num [1:33] 3.57 3.6 3.65 3.75 3.86 ... .$ Var0.03 : num [1:33] 0.00654 0.00816 0.00577 0.00838 0.00953 ... .$ Qt0.04 : num [1:33] 3.6 3.64 3.7 3.79 3.91 ... .$ Var0.04 : num [1:33] 0.00606 0.00664 0.0058 0.00671 0.0068 ... .$ Qt0.05 : num [1:33] 3.63 3.67 3.73 3.83 3.95 ... .$ Var0.05 : num [1:33] 0.00521 0.00618 0.00455 0.00546 0.00602 ... .$ Qt0.06 : num [1:33] 3.66 3.7 3.76 3.86 3.98 ... .$ Var0.06 : num [1:33] 0.00488 0.00522 0.00435 0.00507 0.00572 ... .$ Qt0.07 : num [1:33] 3.69 3.72 3.78 3.89 4.01 ... .$ Var0.07 : num [1:33] 0.00426 0.00464 0.0041 0.00464 0.00512 ... .$ Qt0.08 : num [1:33] 3.71 3.75 3.81 3.92 4.03 ... .$ Var0.08 : num [1:33] 0.00374 0.00408 0.00395 0.00407 0.00477 ... .$ Qt0.09 : num [1:33] 3.74 3.77 3.83 3.94 4.05 ... .$ Var0.09 : num [1:33] 0.00362 0.00377 0.00384 0.00389 0.00462 ... .$ Qt0.1 : num [1:33] 3.75 3.79 3.85 3.96 4.07 ... .$ Var0.1 : num [1:33] 0.00345 0.00324 0.00364 0.00357 0.00441 ... .$ Qt0.11 : num [1:33] 3.77 3.8 3.87 3.98 4.09 ... .$ Var0.11 : num [1:33] 0.00334 0.00335 0.00344 0.00343 0.00457 ... .$ Qt0.12 : num [1:33] 3.79 3.82 3.89 4 4.11 ... .$ Var0.12 : num [1:33] 0.00308 0.00327 0.00314 0.00316 0.00436 ... .$ Qt0.13 : num [1:33] 3.81 3.84 3.9 4.01 4.13 ... .$ Var0.13 : num [1:33] 0.0029 0.00305 0.00305 0.00318 0.00426 ... .$ Qt0.14 : num [1:33] 3.82 3.85 3.92 4.03 4.14 ... .$ Var0.14 : num [1:33] 0.00285 0.00298 0.00286 0.00308 0.00407 ... .$ Qt0.15 : num [1:33] 3.83 3.87 3.93 4.04 4.16 ... .$ Var0.15 : num [1:33] 0.00284 0.00299 0.00289 0.00301 0.00375 ... .$ Qt0.16 : num [1:33] 3.85 3.88 3.95 4.06 4.17 ... .$ Var0.16 : num [1:33] 0.00286 0.00301 0.00271 0.00299 0.00358 ... .$ Qt0.17 : num [1:33] 3.86 3.89 3.96 4.07 4.18 ... .$ Var0.17 : num [1:33] 0.0028 0.00288 0.00254 0.00304 0.00341 ... .$ Qt0.18 : num [1:33] 3.87 3.9 3.97 4.08 4.2 ... .$ Var0.18 : num [1:33] 0.00271 0.00292 0.00251 0.00289 0.00328 ... .$ Qt0.19 : num [1:33] 3.89 3.92 3.99 4.09 4.21 ... .$ Var0.19 : num [1:33] 0.00267 0.00282 0.00244 0.00281 0.00327 ... .$ Qt0.2 : num [1:33] 3.9 3.93 4 4.1 4.22 ... .$ Var0.2 : num [1:33] 0.00258 0.00272 0.00234 0.00283 0.00311 ... .$ Qt0.21 : num [1:33] 3.91 3.94 4.01 4.11 4.24 ... .$ Var0.21 : num [1:33] 0.00255 0.00248 0.00234 0.00273 0.003 ... .$ Qt0.22 : num [1:33] 3.92 3.95 4.02 4.13 4.25 ... .$ Var0.22 : num [1:33] 0.00242 0.0024 0.00228 0.00254 0.00297 ... .$ Qt0.23 : num [1:33] 3.93 3.96 4.03 4.14 4.26 ... .$ Var0.23 : num [1:33] 0.00239 0.00223 0.00219 0.00249 0.00298 ... .$ Qt0.24 : num [1:33] 3.94 3.97 4.04 4.15 4.27 ... .$ Var0.24 : num [1:33] 0.00238 0.00214 0.00214 0.00237 0.00283 ... .$ Qt0.25 : num [1:33] 3.95

3.98 4.05 4.16 4.28 ... .$ Var0.25 : num [1:33] 0.00239 0.00212 0.00213 0.00233 0.00272 ... .$ Qt0.26 : num [1:33] 3.96 3.99 4.06 4.17 4.28 ... .$ Var0.26 : num [1:33] 0.00241 0.00209 0.00211 0.00227 0.00272 ... .$ Qt0.27 : num [1:33] 3.97 3.99 4.07 4.18 4.29 ... .$ Var0.27 : num [1:33] 0.00233 0.00212 0.00211 0.00211 0.00269 ... .$ Qt0.28 : num [1:33] 3.98 4 4.08 4.19 4.3 ... .$ Var0.28 : num [1:33] 0.0023 0.00209 0.00202 0.00214 0.00262 ... .$ Qt0.29 : num [1:33] 3.99 4.01 4.08 4.2 4.31 ... .$ Var0.29 : num [1:33] 0.00223 0.00205 0.00196 0.00208 0.00254 ... .$ Qt0.3 : num [1:33] 4 4.02 4.09 4.2 4.32 ... .$ Var0.3 : num [1:33] 0.00218 0.00195 0.00198 0.00203 0.00245 ... .$ Qt0.31 : num [1:33] 4.01 4.03 4.1 4.21 4.33 ... .$ Var0.31 : num [1:33] 0.00225 0.00192 0.00194 0.00196 0.00235 ... .$ Qt0.32 : num [1:33] 4.02 4.04 4.11 4.22 4.34 ... .$ Var0.32 : num [1:33] 0.00221 0.00195 0.00192 0.00193 0.00235 ... .$ Qt0.33 : num [1:33] 4.03 4.04 4.12 4.23 4.34 ... .$ Var0.33 : num [1:33] 0.00222 0.00193 0.00192 0.00192 0.00233 ... .$ Qt0.34 : num [1:33] 4.03 4.05 4.13 4.24 4.35 ... .$ Var0.34 : num [1:33] 0.00221 0.00195 0.00191 0.00177 0.00223 ... .$ Qt0.35 : num [1:33] 4.04 4.06 4.13 4.24 4.36 ... .$ Var0.35 : num [1:33] 0.00221 0.00198 0.00196 0.00179 0.00216 ... .$ Qt0.36 : num [1:33] 4.05 4.07 4.14 4.25 4.37 ... .$ Var0.36 : num [1:33] 0.00214 0.00198 0.00195 0.00183 0.00218 ... .$ Qt0.37 : num [1:33] 4.06 4.07 4.15 4.26 4.37 ... .$ Var0.37 : num [1:33] 0.0021 0.00201 0.00195 0.00174 0.00212 ... .$ Qt0.38 : num [1:33] 4.07 4.08 4.16 4.27 4.38 ... .$ Var0.38 : num [1:33] 0.00207 0.00203 0.00196 0.0017 0.00209 ... .$ Qt0.39 : num [1:33] 4.07 4.09 4.16 4.27 4.39 ... .$ Var0.39 : num [1:33] 0.00198 0.00194 0.00194 0.00169 0.00207 ... .$ Qt0.4 : num [1:33] 4.08 4.09 4.17 4.28 4.39 ... .$ Var0.4 : num [1:33] 0.00198 0.00191 0.00191 0.00169 0.00207 ... .$ Qt0.41 : num [1:33] 4.09 4.1 4.18 4.29 4.4 ... .$ Var0.41 : num [1:33] 0.00198 0.00187 0.00187 0.00161 0.00199 ... .$ Qt0.42 : num [1:33] 4.09 4.11 4.18 4.29 4.41 ... .$ Var0.42 : num [1:33] 0.00196 0.00185 0.00183 0.00162 0.00199 ... .$ Qt0.43 : num [1:33] 4.1 4.11 4.19 4.3 4.41 ... .$ Var0.43 : num [1:33] 0.00197 0.0018 0.00183 0.00159 0.00192 ... .$ Qt0.44 : num [1:33] 4.11 4.12 4.2 4.31 4.42 ... .$ Var0.44 : num [1:33] 0.002 0.00181 0.00181 0.00159 0.00193 ... .$ Qt0.45 : num [1:33] 4.12 4.12 4.2 4.31 4.43 ... .$ Var0.45 : num [1:33] 0.00195 0.00183 0.00186 0.00163 0.00185 ... .$ Qt0.46 : num [1:33] 4.12 4.13 4.21 4.32 4.43 ... .$ Var0.46 : num [1:33] 0.00192 0.00185 0.0018 0.00156 0.00182 ... .$ Qt0.47 : num [1:33] 4.13 4.14 4.22 4.33 4.44 ... .$ Var0.47 : num [1:33] 0.0019 0.00184 0.00182 0.00155 0.0018 ... .$ Qt0.48 : num [1:33] 4.13 4.14 4.22 4.33 4.45 ... .$ Var0.48 : num [1:33] 0.00186 0.00188 0.00187 0.00157 0.00176 ... .$ Qt0.49 : num [1:33] 4.14 4.15 4.23 4.34 4.45 ... .$ Var0.49 : num [1:33] 0.00186 0.00184 0.00182 0.00151 0.0017 ... .$ Qt0.5 : num [1:33] 4.15 4.15 4.23 4.34 4.46 ... . [list output truncated]

References

MacKinnon, J. G. (2002). Computing numerical distribution functions in econometrics. In proceedings of High Performance Computing Systems and Applications, edited by Pollard, A., Mewhort, D. J. and Weaver, D. F. Springer US. Vol. 451, 455-47

Index