Package 'IAR'

August 28, 2018

Type Package

Title What the package does (short line)

Version 1.0

Date 2018-08-27

Author Who wrote it

Maintainer Who to complain to <yourfault@somewhere.net>

Description More about what it does (maybe more than one line)

License What license is it under?

R topics documented:

IAR-package . . . 2
clcep . . . 3
dscut . . . 3
eb . . . 4
foldlc . . . 4
gentime . . . 5
harmonicfit . . . 6
IAR.gamma . . . 7
IAR.loglik . . . 7
IAR.phi.gamma . . . 9
IAR.phi.loglik . . . 10
IAR.sample . . . 10
IAR.Test . . . 11
IAR.Test2 . . . 12
IARg.sample . . . 13
Planets . . . 14

Index **15**

IAR-package *IAR Model*

Description

This package contains a set of R functions and datasets to perform an autoregressive model to irregularly spaced data

Details

Package:	IAR
Type:	Package
Version:	1.0
Date:	2018-08-27
License:	What license is it under?

~~ An overview of how to use the package, including the most important ~~ ~~ functions ~~

Author(s)

Eyheramendy,S. , Elorrieta, F. and Palma,W. Maintainer: Elorrieta, F. <fielorrieta@mat.puc.cl>

References

Debosscher, J., Sarro, L. M., Aerts, C., Cuypers J., Vandenbussche, B., Garrido, R., & Solano, E. 2007, Astronomy & Astrophysics, 475, 1159.

Elorrieta, F., Eyheramendy, S., Jordn, A., Dekany, I., Catelan, M., Angeloni, R., Alonso-Garcia, J., Contreras-Ramos, R., Gran, F., Hajdu, G., Espinoza, N., Saito, R. & Minniti, D. 2016, Astronomy & Astrophysics, 595, A82.

Jordan, A., Espinoza, N., Rabus, M., Eyheramendy, S., Sing, D.K., Desert, J.M., Bakos, G..,Fortney, J.J., Lopez-Morales, M., Maxted, P.F.L., Triaud, A.,& Szentgyorgyi, A. 2013, The Astrophysical Journal, 778, 184.

Moskalik P. 2014, Proceedings IAU Symposium, 301, 249.

Palma W. 2007, Wiley Series in Probability and Statistics.

Richards, J.W., Starr, D.L., Butler, N.R., Bloom, J.S., Brewer, J.M., Crellin-Quick, A., Higgins,J., Kennedy, R. & Rischard, M. 2011, The Astrophysical Journal, 733, 10.

Zechmeister, M. & Kurster, M. 2009, Astronomy & Astrophysics, 496, 577.

See Also

harmonicfit,gentime,IAR.loglik,IAR.phi.loglik,IAR.sample,IAR.Test,IAR.Test2,IAR.gamma,IARg.sample,fo

clcep *Classical Cepheid*

Description

Time series corresponding to the light curve of a classical cepheid variable star.

Usage

```
data(clcep)
```

Format

A data frame with 109 observations on the following 3 variables.

`t` heliocentric Julian Day

`m` magnitude

`merr` measurement error of the magnitude (in mag).

Details

The frequency computed by GLS for this light curve is 0.060033386

Examples

```
data(clcep)
f1=0.060033386
foldlc(clcep,f1)
```

dscut *Delta Scuti*

Description

Time series corresponding to the light curve of a Delta Scuti variable star.

Usage

```
data(dscut)
```

Format

A data frame with 116 observations on the following 3 variables.

`t` heliocentric Julian Day

`m` magnitude

`merr` measurement error of the magnitude (in mag).

Details

The frequency computed by GLS for this light curve is 14.88558646

Examples

```
data(dscut)
f1=14.88558646
foldlc(dscut,f1)
```

---

eb *Eclipsing Binaries (Beta Lyrae)*

Description

Time series corresponding to the light curve of a Beta Lyrae variable star.

Usage

```
data(eb)
```

Format

A data frame with 470 observations on the following 3 variables.

`t` heliocentric Julian Day

`m` magnitude

`merr` measurement error of the magnitude (in mag).

Details

The frequency computed by GLS for this light curve is 1.510571586

Examples

```
data(eb)
f1=1.510571586
foldlc(eb,f1)
```

---

foldlc *Plotting folded light curves*

Description

This function plotting a time series folded on its period

Usage

```
foldlc(file, f1)
```

Arguments

`file`	Matrix with light curve observations. The first column is the irregular Time, the second column is the brightness magnitude and the third column is the measurement error.
`f1`	Frequency (1/Period) of the raw light curve

Value

A plot of the folded (phased) time series.

Examples

```
data(clcep)
f1=0.060033386
foldlc(clcep,f1)
```

`gentime` *Generating Irregularly spaced times*

Description

Function to generate irregularly spaced times from a mixture of exponential distributions

Usage

```
gentime(n, lambda1 = 130, lambda2 = 6.5, p1 = 0.15, p2 = 0.85)
```

Arguments

`n`	A positive integer. Length of observations times.
`lambda1`	Mean (1/rate) of the first exponential distribution
`lambda2`	Mean (1/rate) of the second exponential distribution
`p1`	Weight of the first exponential distribution
`p2`	Weight of the second exponential distribution

Value

Array with irregularly spaced observations times

See Also

`IAR.sample`

Examples

```
st<-gentime(n=100)
```

harmonicfit *Harmonic Fit to Time Series*

Description

This function fit an k-harmonic function to time series data

Usage

```
harmonicfit(file, f1, nham = 4,weights=NULL,print=FALSE)
```

Arguments

file	A matrix with two columns. The first column corresponds to the observations times, and the second column corresponds to the measures
f1	Frequency (1/Period) of the time series
nham	Number of harmonic components in the model
weights	An array with the weights of each observation
print	logical; if true, the summary of the harmonic fitted model will be printed. The default value is false.

Value

A list with the following components:

res	Residuals to the harmonic fit of the time series.
t	Observations times.
R2	Adjusted R-Squared.
MSE	Mean Squared Error.

Examples

```
data(clcep)
f1=0.060033386
results=harmonicfit(file=clcep[,1:2],f1=f1)
results$R2
results$MSE
results=harmonicfit(file=clcep[,1:2],f1=f1,nham=3)
results$R2
results$MSE
results=harmonicfit(file=clcep[,1:2],f1=f1,weights=clcep[,3])
results$R2
results$MSE
```

IAR.gamma *Maximum Likelihood Estimation of IAR-Gamma model*

Description

Maximum Likelihood Estimation of IAR(1) model

Usage

```
IAR.gamma(y, sT)
```

Arguments

y	Array with the time series observations
sT	Array with the irregular observational times

Examples

```
n=300
set.seed(6714)
st<-gentime(n)
y<-IARg.sample(n,phi=0.9,st,sigma2=1,mu=1)
model<-IAR.gamma(y$y, sT=st)
phi=model$phi
muest=model$mu
sigmaest=model$sigma
```

IAR.loglik *Maximum Likelihood Estimation of IAR(1) model*

Description

Maximum Likelihood Estimation of IAR(1) model

Usage

```
IAR.loglik(y, sT, standarized = "TRUE")
```

Arguments

y	Array with the time series observations
sT	Array with the irregular observational times
standarized	logical; if true, the array y is standarized; if false, y contains the raw time series

Value

A list with the following components:

phi	MLE of the phi coefficient of IAR(1) model.
ll	Value of the negative log likelihood evaluated in phi.

See Also

gentime,IAR.sample,arfima,arima,IAR.phi.loglik

Examples

```
#Generating IAR sample
set.seed(6714)
st<-gentime(n=100)
y<-IAR.sample(phi=0.99,n=100,st)
y<-y$series

#Compute Phi
phi=IAR.loglik(y=y,sT=st)$phi
print(phi)

#Compute the standard deviation of innovations
n=length(y)
d=c(0,diff(st))
phi1=phi**d
yhat=phi1*as.vector(c(0,y[1:(n-1)]))
plot(st,y,type=l)
lines(st,yhat,col=red)
sigma=var(y)
nu=c(sigma,sigma*(1-phi1**(2))[-1])
tau<-nu/sigma
sigmahat<-mean(c((y-yhat)**2/tau))
nuhat<-sigmahat*(1-phi1**(2))
nuhat2<-sqrt(nuhat)

#Equally spaced models
require(arfima)
fit2<-arfima(y,order=c(1,0,0))
fit<-arima(y,order=c(1,0,0),include.mean=FALSE)
syarf<-tacvfARFIMA(phi=fit2$modes[[1]]$phi,dfrac=fit2$modes[[1]]$dfrac,sigma2=fit2$modes[[1]]$sigma,
maxlag=20)[1]
syar<-fit$sigma/(1-fit$coef[1]**2)
print(sigmahat)
print(syar)
print(syarf)

carf<-fit2$modes[[1]]$sigma/syarf
car<-(1-fit$coef[1]**2)
ciar<-(1-phi1**(2))

#Compute the standard deviation of innovations (regular case)

sigma=var(y)
nuhat3=sqrt(sigma*ciar)
searf<-sqrt(sigma*carf)
sear<-sqrt(sigma*car)

#Plot the standard deviation of innovations

plot(st[-1], nuhat3[-1], t="n", axes=FALSE,xlab=Time,ylab=Standard Deviation of Innovations)
axis(1)
axis(2)
```

```
segments(x0=st[-1], y0=nuhat3[-1], y1=0, col=8)
points(st, nuhat3, pch=20, col=1, bg=1)
abline(h=sd(y),col=red,lwd=2)
abline(h=sear,col=blue,lwd=2)
abline(h=searf,col=green,lwd=2)
abline(h=mean(nuhat3[-1]),col=black,lwd=2)
```

`IAR.phi.gamma` *Minus Log Likelihood IAR-Gamma Model*

Description

This function return the negative log likelihood of the IAR-Gamma given a specific values of phi, mu and sigma

Usage

```
IAR.phi.gamma(x, y, sT)
```

Arguments

`x`	An array with the parameters of the IAR-Gamma model. The first element of the array corresponding to the phi parameter, the second to the level parameter mu, and the last one to the scale parameter sigma
`y`	Array with the time series observations
`sT`	Array with the irregular observational times

Value

Value of the negative log likelihood evaluated in phi, mu and sigma.

See Also

`gentime`,`IARg.sample`,`IAR.gamma`,

Examples

```
n=300
set.seed(6714)
st<-gentime(n)
y<-IARg.sample(n,phi=0.9,st,sigma2=1,mu=1)
IAR.phi.gamma(x=c(0.9,1,1),y=y$y,sT=st)
```

IAR.phi.loglik *Minus Log Likelihood IAR(1) Model*

Description

This function return the negative log likelihood of the IAR(1) for a specific value of phi

Usage

```
IAR.phi.loglik(x, y, sT, standarized = "TRUE")
```

Arguments

x	A given phi coefficient of IAR(1) model
y	Array with the time series observations
sT	Array with the irregular observational times
standarized	logical; if true, the array y was standarized; if false, y contains the raw data

Value

Value of the negative log likelihood evaluated in phi.

See Also

gentime,IAR.sample,IAR.loglik,

Examples

```
set.seed(6714)
st<-gentime(n=100)
y<-IAR.sample(phi=0.99,n=100,st)
y<-y$series
IAR.phi.loglik(x=0.8,y=y,sT=st)
```

IAR.sample *Simulate from an IAR(1) Model*

Description

Simulates a IAR(1) Time Series Model

Usage

```
IAR.sample(phi, n = 100, sT)
```

Arguments

phi	A coefficient of IAR(1) model. A value between 0 and 1
n	Length of the output time series. A strictly positive integer.
sT	Array with observational times.

Value

A list with the following components:

times	Array with observation times.
series	Array with simulated IAR(1) data.

See Also

gentime

Examples

```
set.seed(6714)
st<-gentime(n=100)
y<-IAR.sample(phi=0.99,n=100,st)
y<-y$series
plot(st,y,type=l)
```

IAR.Test *Test for the significance of phi*

Description

This function perform a test for the significance of the phi parameter of the IAR model which is based in the residuals of the periodical time series fitted with an harmonic model using an incorrect period.

Usage

```
IAR.Test(y, sT, f, phi, plot = "TRUE", xlim = c(-1, 0))
```

Arguments

y	Array with the time series observations
sT	Array with the irregular observational times
f	Frequency (1/Period) of the raw time series
phi	coefficient phi estimated by IAR.loglik
plot	logical; if true, the function return a density plot of the distribution of the bad fitted examples; if false, this function not return a plot
xlim	The x limits (x1, x2) of the plot. Only works if plot='TRUE'. See plot.default for more details

Details

The null hypothesis of the test is: The phi value corresponds to the computed for the residuals of the data fitted by a wrong model. In this sense, if the hypothesis is rejected, it can be concluded that the residuals of the harmonic model do not remain a time dependency structure. The statistical of the test is log(phi) which was contrasted with a normal distribution with parameters corresponding to the log of the mean and the variance of the phi computed for the residuals of the bad fitted light curves.

Value

A list with the following components:

phi	MLE of the phi coefficient of IAR(1) model.
norm	Mean and variance of the normal distribution of the bad fitted examples.
z0	Statistical of the test (log(phi)).
pvalue	Pvalue computed for Test.

See Also

clcep,harmonicfit,IAR.loglik,IAR.Test2

Examples

```
data(clcep)
f1=0.060033386
results=harmonicfit(file=clcep,f1=f1)
y=results$res/sqrt(var(results$res))
sT=results$t
res3=IAR.loglik(y,sT,standarized=TRUE)[1]
res3$phi
require(ggplot2)
test<-IAR.Test(y=clcep[,2],sT=clcep[,1],f1,res3$phi,plot=TRUE,xlim=c(-10,0.5))
test
```

IAR.Test2 *Test for the significance of phi (Unknown period)*

Description

This function perform a test of the significance of phi which is based in to take N disordered samples of the original data. (Useful for non-periodic time series or when the period is unknown)

Usage

```
IAR.Test2(y, sT, iter = 100, phi, plot = "TRUE", xlim = c(-1, 0))
```

Arguments

y	Array with the values of the time series
sT	Array with the times of the time series
iter	Number of disordered samples of the original data (N)
phi	coefficient phi estimated by IAR.loglik
plot	logical; if true, the function return a density plot of the distribution of the bad fitted examples; if false, this function not return a plot
xlim	The x limits (x1, x2) of the plot. Only works if plot='TRUE'. See plot.default for more details

Details

The main difference regarding to IAR.Test is that to perform this test it is not necessary to know the period of the time series. The null hypothesis of the test is: The phi value corresponds to the computed for the disordered data, which are assumed to be uncorrelated. In this sense, if the hypothesis is accepted, it can be concluded that the observations of the time series are uncorrelated. The statistical of the test is log(phi) which was contrasted with a normal distribution with parameters corresponding to the log of the mean and the variance of the phi computed for the N samples of the disordered data.

Value

A list with the following components:

phi	MLE of the phi coefficient of IAR(1) model.
norm	Mean and variance of the normal distribution of the disordered data.
z0	Statistical of the test (log(phi)).
pvalue	Pvalue computed for Test.

See Also

clcep,harmonicfit,IAR.loglik,IAR.Test

Examples

```
data(Planets)
t<-Planets[,1]
res<-Planets[,2]
y=res/sqrt(var(res))
res3=IAR.loglik(y,t,standarized=TRUE)[1]
res3$phi
set.seed(6713)
require(ggplot2)
test<-IAR.Test2(y=y,sT=t,phi=res3$phi,plot=TRUE,xlim=c(-9.6,-9.45))
```

IARg.sample *Simulate from an IAR-Gamma Model*

Description

Simulates a IAR-Gamma Time Series Model

Usage

```
IARg.sample(n, phi, st, sigma2 = 1, mu = 1)
```

Arguments

n	Length of the output time series. A strictly positive integer.
phi	A coefficient of IAR-Gamma model. A value between 0 and 1
st	Array with observational times.
sigma2	Scale parameter of the IAR-Gamma process. A positive value.
mu	Level parameter of the IAR-Gamma process. A positive value.

Value

A list with the following components:

y	Array with simulated IAR-Gamma process.
st	Array with observation times.

See Also

gentime

Examples

```
n=300
set.seed(6714)
st<-gentime(n)
y<-IARg.sample(n,phi=0.9,st,sigma2=1,mu=1)
plot(st,y$y,type=l)
hist(y$y,breaks=20)
```

Planets *Planets*

Description

Time series corresponding to the residuals of the parametric model fitted by Jordan et al (2013) for a transit of an extrasolar planets

Usage

```
data(Planets)
```

Format

A data frame with 91 observations on the following 2 variables.

`t` Time from mid-transit (hours)

`r` Residuals of the parametric model fitted by Jordan et al (2013)

Source

Jordan, A., Espinoza, N., Rabus, M., Eyheramendy, S., Sing, D.K., Desert, J.M., Bakos, G..,Fortney, J.J., Lopez-Morales, M., Maxted, P.F.L., Triaud, A.,& Szentgyorgyi, A. 2013, The Astrophysical Journal, 778, 184.

Examples

```
data(Planets)
plot(Planets[,1],Planets[,2],xlab=Time from mid-transit (hours),ylab=Noise,pch=20)
```

Index

∗Topic **\textasciitildekwd1**
foldlc, 4
gentime, 5
harmonicfit, 6
IAR.gamma, 7
IAR.loglik, 7
IAR.phi.gamma, 9
IAR.phi.loglik, 10
IAR.sample, 10
IAR.Test, 11
IAR.Test2, 12
IARg.sample, 13
∗Topic **\textasciitildekwd2**
foldlc, 4
gentime, 5
IAR.gamma, 7
IAR.loglik, 7
IAR.phi.gamma, 9
IAR.phi.loglik, 10
IAR.sample, 10
IAR.Test, 11
IAR.Test2, 12
IARg.sample, 13
∗Topic **autoregressive, harmonic, irregular time series**
IAR-package, 2
∗Topic **datasets**
clcep, 3
dscut, 3
eb, 4
Planets, 14

arfima, *8*
arima, *8*

clcep, 3, *12, 13*

dscut, 3

eb, 4

foldlc, *2*, 4

gentime, *2*, 5, *8–11*, *14*
ggplot, *2*

harmonicfit, *2*, 6, *12, 13*

IAR *(*IAR-package*)*, 2
IAR-package, 2
IAR.gamma, *2*, 7, *9*
IAR.loglik, *2*, 7, *10–13*
IAR.phi.gamma, 9
IAR.phi.loglik, *2*, *8*, 10
IAR.sample, *2*, *5*, *8*, *10*, 10
IAR.Test, *2*, 11, *13*
IAR.Test2, *2*, *12*, 12
IARg.sample, *2*, *9*, 13

Planets, 14
plot.default, *11, 12*