Package ‘AITS’

June 28, 2019

Type Package

Title What the package does (short line)

Version 1.0

Date 2019-06-28

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:

`AITS-package` *Analysis of Irregularly spaced time series*

Description

This package contains a set of R functions and datasets to perform both the Irregular Autoregressive Model (IAR model) and the Complex Irregular Autoregressive Model (CIAR model) which are useful for unequally spaced time series.

Details

Package:	AITS
Type:	Package
Version:	2.0
Date:	2017-02-01
License:	What license is it under?

The package AITS contains R functions to fit unequally spaced time series from the Irregular Autoregressive (IAR) and the Complex Irregular Autoregressive (CIAR) models. From the functions of this package we can generate observations for each process, compute the negative of the log likelihood of these process, fit each model to irregularly sampled data, and test the significance of the estimated parameters. The examples are focused in astronomical data, due to the large number of unequally spaced time series which can be found in this field. Particularly, we expect that this model be useful in the analysis of variable stars and extrasolar planets. Furthermore, we add complementary functions to analyze this model. For example, a function to generate irregular times from the mixture of exponential distributions, other to plot a light curve folded on its period and the last one to fit an harmonic model are also included in this package.

Author(s)

Elorrieta, F., Eyheramendy, S. 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., Jordan, 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.

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487. URL http://dx.doi.org/10.1093/mnras/sty2487.

Eyheramendy, S., Elorrieta, F., Palma, W. 2016, An autoregressive model for irregular time series of variable stars. Proceedings of the International Astronomical Union, 12(S325):259-262, 2016. doi: 10.1017/S1743921317000448.

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.

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.kalman`,`IAR.phi.kalman`,`IAR.sample`,`IAR.Test`,`IAR.Test`

`agn` *Active Galactic Nuclei*

Description

Time series corresponding to the AGN MCG-6-30-15 light curve measured in the K-band

Usage

```
data(agn)
```

Format

A data frame with 237 observations on the following 3 variables.

`t` heliocentric Julian Day - 2450000

`m` Flux (10^(-15) ergs/s/cm^2 /A)

`merr` measurement error standard deviations.

References

Lira, P., Arévalo, P., Uttley, P., McHardy, I. M. M., & Videla, L. 2015, Monthly Notices of the Royal Astronomical Society, 454, 368

Examples

```
data(agn)
plot(agn$t,agn$m,type="l",ylab="",xlab="")
```

CIAR.kalman *Maximum Likelihood Estimation of the CIAR Model*

Description

Maximum Likelihood Estimation of the CIAR model parameters phi.R and phi.I. The estimation procedure uses the Kalman Filter to find the maximum of the likelihood

Usage

```
CIAR.kalman(y, t, delta = 0, zero.mean = "TRUE", standarized = "TRUE", c = 1, niter = 10, seed = 1234)
```

Arguments

y	Array with the time series observations
t	Array with the irregular observational times
delta	Array with the measurements error standard deviations
zero.mean	logical; if true, the array y has zero mean; if false, y has a mean different from zero.
standarized	logical; if true, the array y is standarized; if false, y contains the raw time series
c	Nuisance parameter corresponding to the variance of the imaginary part.
niter	Number of iterations in which the function nlminb will be repeated
seed	a single value, interpreted as the seed of the random process.

Value

A list with the following components:

phiR	MLE of the Real part of the coefficient of CIAR model.
phiI	MLE of the Imaginary part of the coefficient of the CIAR model.
ll	Value of the negative log likelihood evaluated in phiR and phiI.

See Also

gentime,CIAR.sample,CIAR.phi.kalman,IAR.kalman,dmcep,agn

Examples

```
n=300
set.seed(6714)
st<-gentime(n)
x=CIAR.sample(n=n,phi.R=0.9,phi.I=0,sT=st,c=1)
options(digits=4)
y=x$y
y1=y/sd(y)
ciar=CIAR.kalman(y=y1,t=st)
ciar
Mod(complex(real=ciar$phiR,imaginary=ciar$phiI))

#Detecting Negative Autocorrelation
```

```
data(dmcep)
f1=0.7410152
foldlc(dmcep,f1)
fit=harmonicfit(dmcep,f1)
y<-fit$res
y1=y/sqrt(var(y))
res3=CIAR.kalman(y1,dmcep$t,c=1)
print(res3$phiR)
phi=IAR.loglik(y1,dmcep$t)
print(phi$phi)

#CIAR in Non-Normalized Data
data(agn)
st<-agn[,1]
y<-agn[,2]
yerr<-agn[,3]
y0=(y-mean(y))/sqrt(var(y))
y1=y-mean(y)

#Normalized Data
ciar=CIAR.kalman(y=y0,t=st,delta=yerr/sqrt(var(y)),seed=1234)
ciar$phiR
iar=IAR.kalman(y=y0,sT=st,delta=yerr/sqrt(var(y)))
iar$phi

#Zero Mean Data
ciar=CIAR.kalman(y=y1,t=st,delta=yerr,standarized="FALSE",seed=1234)
ciar$phiR
iar=IAR.kalman(y=y1,sT=st,delta=yerr,standarized="FALSE")
iar$phi


#Original Data
ciar=CIAR.kalman(y=y,t=st,delta=yerr,zero.mean="FALSE",standarized="FALSE",seed=1234)
ciar$phiR
iar=IAR.kalman(y=y,sT=st,delta=yerr,zero.mean="FALSE",standarized="FALSE")
iar$phi
```

`CIAR.phi.kalman` *Minus Log Likelihood of the CIAR Model*

Description

This function return the negative log likelihood of the CIAR process given specific values of phi.R and phi.I

Usage

```
CIAR.phi.kalman(x, y, t, yerr, zero.mean = "TRUE", standarized = "TRUE", c = 1)
```

Arguments

x	An array with the parameters of the CIAR model. The elements of the array are, in order, the real and the imaginary part of the phi parameter of the CIAR model.
y	Array with the time series observations
t	Array with the irregular observational times
yerr	Array with the measurements error standard deviations
zero.mean	logical; if true, the array y has zero mean; if false, y has a mean different from zero.
standarized	logical; if true, the array y is standarized; if false, y contains the raw time series
c	Nuisance parameter corresponding to the variance of the imaginary part.

Value

Value of the negative log likelihood evaluated in phiR and phiI.

Examples

```
n=300
set.seed(6714)
st<-gentime(n)
x=CIAR.sample(n=n,phi.R=0.9,phi.I=0,sT=st,c=1)
y=x$y
yerr=rep(0,n)
CIAR.phi.kalman(x=c(0.8,0),y=y,t=st,yerr=yerr)
```

CIAR.sample *Simulate from an CIAR Model*

Description

Simulates a CIAR Time Series Model

Usage

```
CIAR.sample(n, sT, phi.R, phi.I, rho = 0, c = 1)
```

Arguments

n	Length of the output time series. A strictly positive integer.
sT	Array with observational times.
phi.R	Real part of the coefficient of CIAR model. A value between -1 and 1.
phi.I	Imaginary part of the coefficient of CIAR model. A value between -1 and 1.
rho	Correlation between the real and the imaginary part of the process. A value between -1 and 1.
c	Nuisance parameter corresponding to the variance of the imaginary part.

Details

The values phi.R and phi.I must be chosen in order to satisfy the condition |phi.R + i phi.I| < 1.

Value

A list with the following components:

y	Array with simulated IAR-Gamma process.
t	Array with observation times.
Sigma	Covariance matrix of the process.

See Also

gentime

Examples

```
n=300
set.seed(6714)
st<-gentime(n)
x=CIAR.sample(n=n,phi.R=0.9,phi.I=0,sT=st,c=1)
plot(st,x$y,type='l')
```

CIAR.Test *Test for the significance of the real part of phi*

Description

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

Usage

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

Arguments

y	Array with the time series observations
sT	Array with the irregular observational times
c	Nuisance parameter corresponding to the variance of the imaginary part.
f	Frequency (1/Period) of the raw time series
phi	parameter phi^R estimated by CIAR.kalman
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
Mod	logical; if true, the test is performed using the module of the complex coefficient phi; if false, this function is performed using the real part of the complex coefficient, ie, phi^R. The default is FALSE.

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 phiR coefficient (or the Module of the complex coefficient phi) of CIAR model.
`norm`	Mean and variance of the normal distribution of the bad fitted examples.
`z0`	Statistical of the test (log(|phiR|)).
`pvalue`	Pvalue computed for the Test.

See Also

`clcep`,`harmonicfit`,`CIAR.kalman`,`IAR.Test`

Examples

```
data(clcep)
f1=0.060033386
results=harmonicfit(file=clcep,f1=f1)
y=results$res/sqrt(var(results$res))
sT=results$t
res3=CIAR.kalman(y,sT,standarized='TRUE')
res3$phiR
require(ggplot2)
test<-CIAR.Test(y=clcep[,2],sT=clcep[,1],f=f1,phi=res3$phiR,plot='TRUE',xlim=c(-37.5,0.5))
test
```

`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)
```

---

dmcep *Double Mode Cepheid*

Description

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

Usage

```
data(dmcep)
```

Format

A data frame with 191 observations on the following 3 variables.

`t` heliocentric Julian Day

`m` magnitude

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

Details

The dominant frequency computed by GLS for this light curve is 0.7410152. The second frequency computed by GLS for this light curve is 0.5433353.

Examples

```
data(dmcep)
f1=0.7410152
foldlc(dmcep,f1)
fit=harmonicfit(dmcep,f1)
f2=0.5433353
foldlc(cbind(dmcep$t,fit$res,dmcep$merr),f2)
```

`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)
```

---

forecast.CIAR *Forecast from models fitted by* CIAR.kalman

Usage

```
forecast.CIAR(phi.R, phi.I, y1, st, n.ahead = 1)
```

Arguments

phi.R	Real part of the phi coefficient of CIAR model.
phi.I	Imaginary part of the phi coefficient of CIAR model.
y1	Array with the time series observations
st	Array with the irregular observational times
n.ahead	The number of steps ahead for forecast is required.

See Also

CIAR.kalman,predict.CIAR,agn

Examples

```
#Simulated Data
n=300
set.seed(6714)
st<-gentime(n)
x=CIAR.sample(n=n,phi.R=0.9,phi.I=0,sT=st,c=1)
options(digits=4)
y=x$y
y1=y/sd(y)
n=length(y1)
p=trunc(n*0.90)
ytr=y1[1:p]
yte=y1[(p+1):n]
str=st[1:p]
ste=st[(p+1):n]
n.ahead=ste-str[p]

final<-matrix(0,length(n.ahead),4)
for(i in 1:length(n.ahead))
{
ciar=CIAR.kalman(y=ytr,t=str)
ciar
n.ahead=ste[i]-str[length(str)]
forCIAR<-forecast.CIAR(ciar$phiR,ciar$phiI,ytr,str,n.ahead=n.ahead)
final[i,]<-c(ciar$phiR,yte[i],forCIAR$forecast,n.ahead)
ytr=c(ytr,yte[i])
str=c(str,ste[i])
}

plot(st,seq(from=min(y1),to=max(y1),length.out=length(st)),type="n",xlim=c(0,max(st)),ylab="")
lines(st,y1)
points(ste,final[,3],col="red",pch=20)
points(ste,final[,2],col="black",pch=20)
segments(x0=ste, y0=final[,2], y1=final[,3], col=8)

#AGN Data
data(agn)
st<-agn[,1]
y<-agn[,2]
y1=(y-mean(y))/sd(y)
n=length(y1)
p=trunc(n*0.90)
ytr=y1[1:p]
yte=y1[(p+1):n]
str=st[1:p]
ste=st[(p+1):n]
n.ahead=ste-str[p]

final<-matrix(0,length(n.ahead),4)
for(i in 1:length(n.ahead))
{
ciar=CIAR.kalman(y=ytr,t=str)
```

```
ciar
n.ahead=ste[i]-str[length(str)]
forCIAR<-forecast.CIAR(ciar$phiR,ciar$phiI,ytr,str,n.ahead=n.ahead)
final[i,]<-c(ciar$phiR,yte[i],forCIAR$forecast,n.ahead)
ytr=c(ytr,yte[i])
str=c(str,ste[i])
}

plot(st,seq(from=min(y1),to=max(y1),length.out=length(st)),type="n",xlim=c(min(st),max(st)),ylab="")
lines(st,y1)
points(ste,final[,3],col="red",pch=20)
points(ste,final[,2],col="black",pch=20)
segments(x0=ste, y0=final[,2], y1=final[,3], col=8)
```

---

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

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

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 the IAR-Gamma model*

Description

Maximum Likelihood Estimation of the IAR-Gamma model

Usage

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

Arguments

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

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

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.kalman` *Maximum Likelihood Estimation of the IAR Model with known measurement error variances*

Description

Maximum Likelihood Estimation of the IAR model parameter phi. The estimation procedure uses the Kalman Filter to find the maximum of the likelihood

Usage

```
IAR.kalman(y, sT, delta = 0, zero.mean = "TRUE", standarized = "TRUE")
```

Arguments

`y`	Array with the time series observations
`sT`	Array with the irregular observational times
`delta`	Array with the measurements error standard deviations
`zero.mean`	logical; if true, the array y has zero mean; if false, y has a mean different from zero.
`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 the IAR model.
`ll`	Value of the negative log likelihood evaluated in phi.

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

See Also

`gentime`,`IAR.sample`,`IAR.phi.kalman`

Examples

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

`IAR.loglik` *Maximum Likelihood Estimation of the IAR Model*

Description

Maximum Likelihood Estimation of the IAR 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 the IAR model.
ll	Value of the negative log likelihood evaluated in phi.

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

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 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.

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

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.kalman` *Minus Log Likelihood of the IAR Model with known measurement error variances*

Description

This function return the negative log likelihood of the IAR process given a specific value of phi

Usage

```
IAR.phi.kalman(x, y, yerr, t, zero.mean = "TRUE", standarized = "TRUE")
```

Arguments

x	
y	Array with the time series observations
yerr	Array with the measurements error standard deviations
t	Array with the irregular observational times
zero.mean	logical; if true, the array y has zero mean; if false, y has a mean different from zero.
standarized	logical; if true, the array y is standarized; if false, y contains the raw time series

Value

Value of the negative log likelihood evaluated in phi.

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

See Also

gentime,IAR.sample,IAR.kalman,

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.phi.loglik` *Minus Log Likelihood of the IAR Model*

Description

This function return the negative log likelihood of the IAR Model for a specific value of phi

Usage

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

Arguments

`x`	A given phi coefficient of the IAR 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.

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

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.phi.t` *Minus Log Likelihood IAR-T Model*

Description

This function return the negative log likelihood of the IAR-T given specific values of phi and sigma

Usage

```
IAR.phi.t(x, y, sT, nu = 3)
```

Arguments

`x`	An array with the parameters of the IAR-T model. The first element of the array corresponding to the phi parameter and the second element to the scale parameter sigma
`y`	Array with the time series observations
`sT`	Array with the irregular observational times
`nu`	degrees of freedom

Value

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

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

See Also

`gentime`,`IARt.sample`,`IAR.t`,

Examples

```
n=300
set.seed(6714)
st<-gentime(n) #Unequally spaced times
y<-IARt.sample(n,0.9,st,sigma2=1,nu=3)
IAR.phi.t(x=c(0.9,1),y=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.

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

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.t *Maximum Likelihood Estimation of the IAR-T model*

Description

Maximum Likelihood Estimation of the IAR-T model

Usage

```
IAR.t(y, sT, nu = 3)
```

Arguments

y	Array with the time series observations
sT	Array with the irregular observational times
nu	degrees of freedom

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

Examples

```
n=300
set.seed(6714)
st<-gentime(n)
y<-IARt.sample(n,0.9,st,sigma2=1,nu=3)
model<-IAR.t(y$y, sT=st)
phi=model$phi
sigmaest=model$sigma
```

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 the IAR 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.

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

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 the IAR 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.

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

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 an 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.

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

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)
```

`IARt.sample` *Simulate from an IAR-T Model*

Description

Simulates an IAR-T Time Series Model

Usage

```
IARt.sample(n, phi, st, sigma2 = 1, nu = 3)
```

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.
`nu`	degrees of freedom

Value

A list with the following components:

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

References

Eyheramendy, S., Elorrieta, F., Palma, W. 2018, An irregular discrete time series model to identify residuals with autocorrelation in astronomical light curves. Monthly Notices of the Royal Astronomical Society, 481(4):4311–4322, 2018. doi: 10.1093/mnras/sty2487.

See Also

`gentime`

Examples

```
n=300
set.seed(6714)
st<-gentime(n)
y<-IARt.sample(n,0.9,st,sigma2=1,nu=3)
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)

References

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)
```

predict.CIAR	*Fitted Values of CIAR model*

Usage

```
predict.CIAR(x, y, t, standarized = "TRUE", c = 1)
```

Arguments

`x`	An array with the parameters of the CIAR model. The elements of the array are, in order, the real and the imaginary part of the phi parameter of the CIAR model.
`y`	Array with the time series observations
`t`	Array with the irregular observational times
`standarized`	logical; if true, the array y is standarized; if false, y contains the raw time series
`c`	Nuisance parameter corresponding to the variance of the imaginary part.

See Also

`gentime`,`CIAR.sample`,`CIAR.phi.kalman`,`forecast.CIAR`

Examples

```
n=300
set.seed(6714)
st<-gentime(n)
x=CIAR.sample(n=n,phi.R=0.9,phi.I=0,sT=st,c=1)
options(digits=4)
y=x$y
y1=y/sd(y)
ciar=CIAR.kalman(y=y1,t=st)
ciar
yhat=predict.CIAR(x=c(ciar$phiR,ciar$phiI),y=y1,t=st)
```

Index