Chapter 1

Functions Implemented for IAR model

In this document, the functions developed in the freeware R (R Core Team 2015 [2]) to implement the IAR model are described.

In total there are eleven functions, five of them for the IAR process and three for the IAR-Gamma Process. The three remaining functions are for generating the irregular times from the mixture of exponential distributions (equation **??**), fitting an harmonic model (equation **??**) and plotting the folded light curve. The functions implemented for the irregular time series models allow to generate observations for each process, fit each model and test the significance of the autocorrelation parameter according with the two tests.

In addition, four time series are added to test the functions. Three of these time series corresponds to light curves of a Classical Cepheid (`clcep`), Delta Scuti (`dscut`) and a Beta Lyrae (`eb`) eclipsing binaries variable star. In the R documentation of each light curve, the frequency estimated by GLS was added. The remaining time series corresponds to the residuals of the fitted exoplanet transit light curve (`Planets`) presented in the section **??**.

1.1 The gentime function

The first function allows to generate synthetic irregular time series. To do this, first the irregular times must be generated. We propose to generate the irregular times using a mixture of exponential distributions (equation **??**). The function `gentime` allows to simulate the irregular times. The function can be implemented by the following R command.

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

where `n` corresponds to the number of observational times that will be generated. `lambda1` and `lambda2` are the means of each exponential distribution, `p1` and `p2` are its respective weights. The result of this function is an array with the irregularly spaced observations times.

1.2 The harmonicfit function

It is well-known that the light curves of variable stars have a periodical behavior. If the period of a specific variable star is known, this light curve can be fitted by an harmonic model. From the function `harmonicfit` this model can be fitted to an irregular time series, using the following R command,

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

where `file` is a matrix with two columns, the first of them must have the irregular times and the second the observations. Furthermore, `f1` is the frequency of the time series that will be modeled and `nham` is the number of harmonics that can be used in the fit. The default is 4. In addition, to fit a weighted harmonic model, an array with the weights of each observation must be added in the argument `weights`. Finally, the `print` argument is a boolean. When `print = TRUE` a summary of the harmonic model fitted will be printed. The data `clcep` can be used to test this function. An harmonic model can be fitted to this time series using the follow command,

```
data(clcep)
f1=0.060033386
results=harmonicfit(file=clcep,f1=f1,nham=4)
```

The function `harmonicfit` returns both the residuals of the fitted model and goodness of fit measures, such as the R squared (R2) and the Mean Squared Error (MSE).

1.3 The foldlc function

The light curves of variable stars that have a periodical behavior are generally plotted in its phase. In the phased light curve, the periodic behavior of the brightness a star can be seen much better than in the irregularly measured raw light curve. In equation (**??**) the phase of an observation ϕ is defined. The phased light curve also is currently called folded light curve. To make the plot of the folded (phased) light curve with this package, the following code must be used,

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

where `file` is a matrix with three columns, corresponding to the irregular times, the magnitudes and the measurement errors, and the `f1` is the frequency of the light curve.

The three functions explained above are incorporated in order to facilitate the application of the irregular time series models in the light curves of variable stars. The remaining functions are useful to simulate and modeling the irregular time series process. First, the functions that allow to simulate each process are described below.

1.4 Simulating the Irregular Time Series Processes

The functions `IAR.sample` and `IARg.sample` allow to generate observations from the IAR and IAR-Gamma processes respectively. All these functions work similarly, in the sense that they need as input the length of the generated time series (`n`), the vector with the irregular times (`sT`), which can be generated using the function `gentime`, and the specific parameters of each model. For example, to generate an IAR process the following R command must be used,

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

where `phi` is the value of the autocorrelation parameter of the simulated data. `phi` can take values in the interval (0,1). The R-command to simulate an IAR-Gamma process is,

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

as can be seen, the last two commands are very similar. However, the IAR gamma needs to specify two additional parameters according with equation **??**, the scale parameter `sigma2` and the level parameter `mu`.

As an example of the use of these functions, the following script generate a IAR process of length 300 with irregular times coming from the mixture of exponential distributions, with parameters $\lambda_1 = 130$, $\lambda_2 = 6.5$, $\omega_1 = 0.15$ and $\omega_2 = 0.85$, and the time dependency parameter $\phi = 0.9$.

```
set.seed(6714)
st<-gentime(n=300)
y<-IAR.sample(phi=0.9,n=300,st)
y<-y$series
plot(st,y,type='l')
```

In Figure 1.1, the simulated IAR process are shown. It can be observed a stationary behavior of its mean and variance.

1.5 Fitting the Irregular Time Series Processes

The estimation of the two models implemented can be performed by maximum likelihood. However, the estimators of the autocorrelation parameters do not have a closed form, whereby iterative methods must be used. These iterative methods have already been implemented in other packages, for example, the `optimize` function of the `stats` package allows us to find the optimal estimator of the ϕ parameter of the IAR model. However, for the IAR Gamma it is necessary to optimize more than one parameter. Consequently, the function `nlminb` of the `stats` package allows us to find the optimal solution for these models.

To estimate the IAR model parameter ϕ, the function `IAR.loglik` must be implemented by,

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

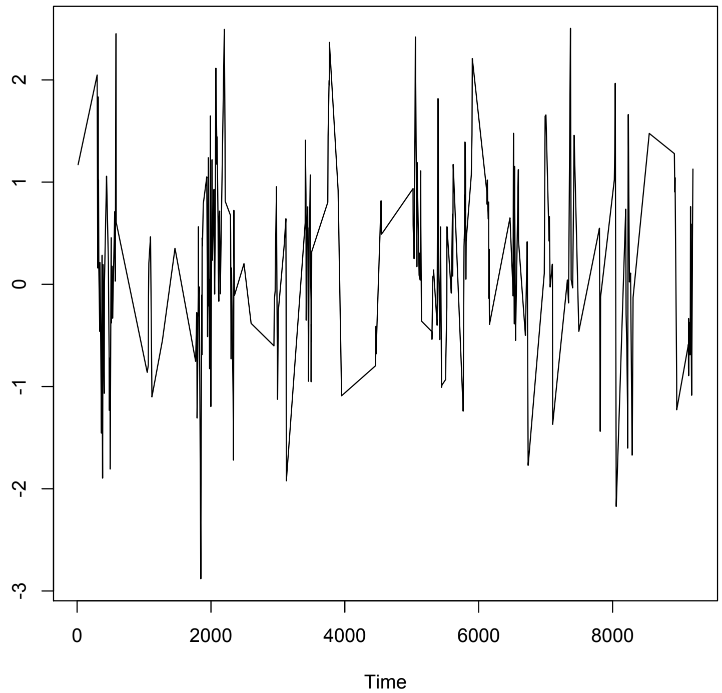

Figure 1.1: Simulated IAR Time Series of length 300 and $\phi = 0.9$. The times was generated by the mixture of exponential distributions with parameters $\lambda_1 = 130$, $\lambda_2 = 6.5$, $\omega_1 = 0.15$ and $\omega_2 = 0.85$.

where `y` is the array with the values of the sequence, `sT` is the array with the irregular times and the boolean `standarized` must be "TRUE" if the array `y` was standarized. As for example, the ϕ parameter of the IAR process shown in Figure 1.1 will be estimated with the following code.

```
set.seed(6714)
st<-gentime(n=300)
y<-IAR.sample(phi=0.9,n=300,st)
y<-y$series
phi=IAR.loglik(y=y,sT=st)$phi
phi
[1] 0.898135
```

Note that the estimated value was $\hat{\phi} = 0.898$, which was very close to the value with which the IAR process was generated ($\phi = 0.9$). The fitted values of the IAR process and the maximum likelihood estimator of σ^2 can be obtained with the following code,

```
n=300
```

```
d=c(0,diff(st))
phi1=phi**d
yhat=phi1*as.vector(c(0,y[1:(n-1)]))
sigma=var(y)
nu=c(sigma,sigma*(1-phi1**(2))[-1])
tau<-nu/sigma
var.hat=mean(((y-yhat)**2)/tau)
var.hat
[1] 0.9506582
```

where `yhat` is the vector of the fitted values and `var.hat` is the maximum likelihood estimation of the variance of the process.

Similarly to the IAR process, the function `IAR.gamma` allows to estimate the parameters of the IAR-Gamma process using the following command,

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

where `y` is the array with the values of the sequence and `sT` is the array with the irregular times. In order to test this function, a IAR.gamma process will be generated with the following code,

```
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)
```

In Figure 1.2 a) the IAR.gamma time series is shown. In order to show the asymmetrical behavior of this time series, the histogram of the IAR.gamma observations has been added in figure b).

The maximum likelihood estimation of the IAR-Gamma parameters ϕ, μ y σ (see equation **??**) can be performed on the simulated time series with the following R-command,

```
model<-IAR.gamma(y$y, sT=st)
phi=model$phi
muest=model$mu
sigmaest=model$sigma
phi
[1] 0.8990846
```

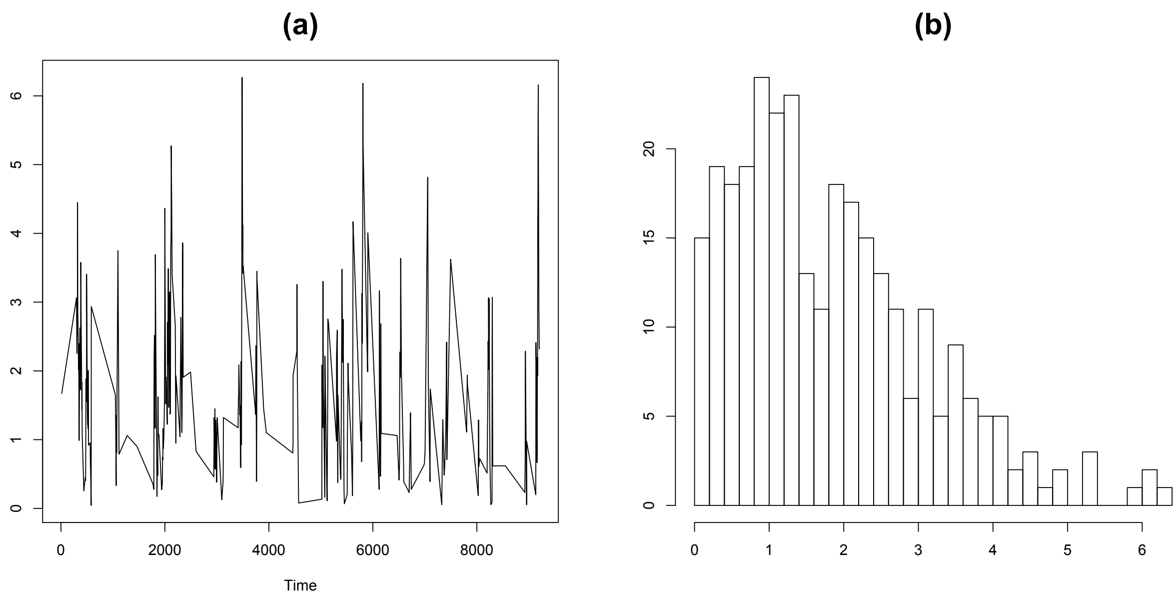

Figure 1.2: Figure a) shows the time series of the Simulated IAR-Gamma Process with length 300 and $\phi = 0.9$. The times was generated by the mixture of exponential distributions with parameters $\lambda_1 = 130$, $\lambda_2 = 6.5$, $\omega_1 = 0.15$ and $\omega_2 = 0.85$. Figure (b) shows the histogram of the IAR-Gamma observations

```
muest
[1] 0.9349599
sigmaest
[1] 0.9854233
```

Note that the estimation of the three parameters $\hat{\phi}$, $\hat{\mu}$, $\hat{\sigma}$ of the IAR-gamma process was very accurate, taking values 0.899, 0.934 and 0.985 respectively.

1.6 Testing the significance of the parameters of the irregular models

The construction of two statistical test to assess the significance of the autocorrelation parameters was shown previously. The formulated tests differ in that the first of them assumes that the time series have a periodical behavior which can be modeled by an harmonic model. The main idea of this test is to verify whether the harmonic model explain all the time dependency structure in the time series or not. If not, a time dependency structure should remain in the residual of the harmonic fit.

To assess the significance of the autocorrelation parameter, this test uses the dominant frequency (which can be found by GLS (**??**)). This frequency is used to fit an harmonic model to the time series. Later, the residuals of the harmonic fitted model are modeled by the irregular time series. The parameter estimated ϕ can be used as an autocorrelation index. However, as mentioned previously a small value of $\hat{\phi}$ does not necessarily mean that there is no temporal dependence on the time series, since this may be due to the dependence between the frequency and the phi value discussed in section **??**. To verify whether the residuals are uncorrelated or not, the test fit an harmonic model to the raw time series using now a percentual variation of the correct frequency. As these models are fitted using a wrong period, $\hat{\phi}$ must have greater values regarding to the ones obtained in the residuals of the correct fitted model.

To perform this test for the ϕ parameter estimated by the IAR model, the following command must be used,

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

where `y` is the array with the time series observations and `sT` is the array with the irregular observational times. In addition, the dominant frequency `f` and the ϕ parameter estimated by the IAR model (`IAR.loglik`) are needed as input. The argument `plot` is logical, if it is true, the function returns a density plot of the distribution of the $\hat{\phi}$ estimated in the residuals of the wrongly fitted models. The argument `xlim` only works if `plot = "TRUE"`, and define the limits of the x axis. The data `clcep` of this package can be used to exemplify the use of this function. With the following code the example of this test for the IAR model can be run,

```
data(clcep)
f1=0.060033386
results=harmonicfit(file=clcep,f1=f1)
y=results$res/sqrt(var(results$res))
sT=results$t
res3=IAR.loglik(y,sT,standarized='TRUE')
require(ggplot2)
test<-IAR.Test(y=clcep[,2],sT=clcep[,1],f1,res3$phi,plot='TRUE',xlim=c(-10,0.5))
test
```

In this example the ϕ estimated by the IAR model is $\hat{\phi} = 6.67e-05$ and the p-value of the test is 0. According with the hypothesis of the test defined in the section **??**, the ϕ estimated value is not significative. Therefore, the residuals of the harmonic fit do not have a time dependency structure. In Figure 1.3 a) it is shown the density plot returned for the function `IAR.Test`. This plot has the density of the $log(\phi)$ estimated in the residuals when the time series was fitted wrongly, and the red triangle is the $log(\phi)$ of the "correct" ϕ estimation. Evidently, this point does not belong to the distribution of the "wrong" estimations.

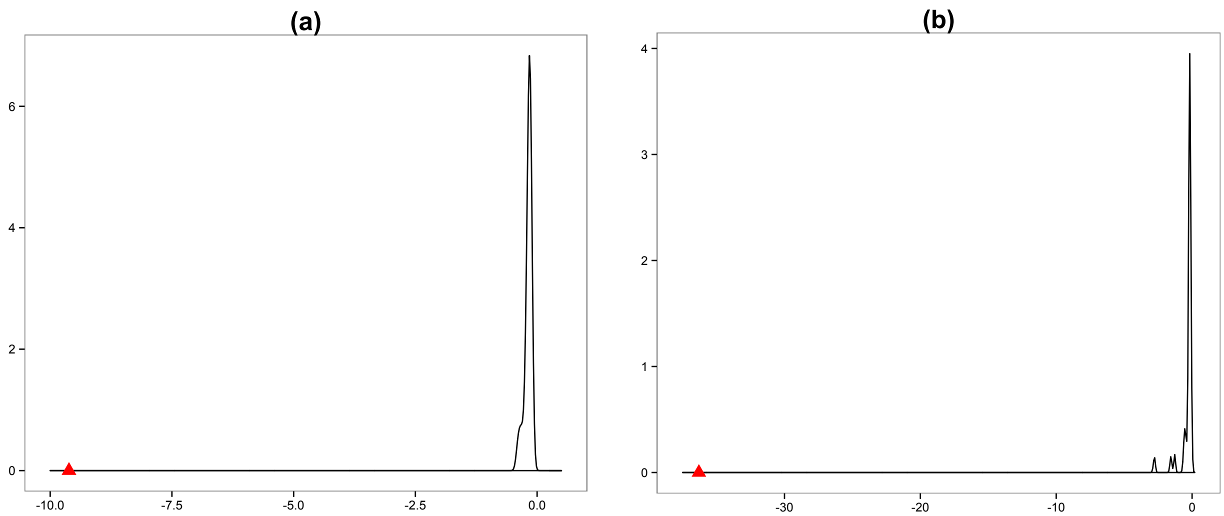

Figure 1.3: Figure a) shows the Density Plot of the log(ϕ), where ϕ was estimated by the IAR model when the time series is fitted using wrong periods. The red triangle is the log(ϕ) estimated when the correct period is used in the harmonic fit. Figure (b) shows the same plot for the CIAR process.

As mentioned previously, in this package there are two different kind of test. The second one, do not assumes a periodical behavior of the time series. Therefore, this time series is not able to fit an harmonic model. The main idea of this test is shuffling the time series several times, generating in each case independent samples of the original time series or breaking the time dependency. The ϕ estimated in the "independent" time series must be less than the ϕ estimated in the raw time series. To perform this test, the following code must be used,

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

This function works with the same arguments as the `IAR.Test` function, with the exception that it does not require knowing the frequency of the time series. Instead, the function `IAR.Test2` uses the argument `iter` to define the number of independent time series that will be used to create the distribution of log(ϕ). In order to exemplify the use of this function, the code used in the application on the light curve of an exoplanet transit described in section **??** will be shown below,

```
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))
test
```

The data `Planets` corresponds to the residuals of the fitted model by Jordan et al,2013 [1] in an exoplanet transit. In this example, the p-value was ≈ 1, therefore the null hypothesis was accepted. Consequently, it is confirmed that in the Planets time series there is no structure of temporal correlation.

Bibliography

[1] A. Jordán, N. Espinoza, M. Rabus, S. Eyheramendy, D. K. Sing, J.-M. Désert, G. Á. Bakos, J. J. Fortney, M. López-Morales, P. F. L. Maxted, A. H. M. J. Triaud, and A. Szentgyorgyi. A Ground-based Optical Transmission Spectrum of WASP-6b. , 778:184, December 2013. doi: 10.1088/0004-637X/778/2/184.

[2] R Core Team. R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, 2015.