

Séries temporelles

Séances de TP du 07 et 14 mars 2016

Pour ce qui suit installer et charger les packages suivants :

```
library(stats)
library(tseries)
library(RColorBrewer)
source('~/Documents/ENSEIGNEMENT/EVRY/ENS15-16/Séries temporelles M1/arma.scan.R')
#library(astsa)
#library(forecast)
#library(timeSeries)
#library(fArma)
#library(fGarch)
```

Exercice 1

Dans ce qui suit w_t est un bruit blanc gaussien standard.

1. Simulation de processus autorégressif (AR) :

- (a) Soit w_t un bruit blanc gaussien standard. Simuler un processus AR(1) $x_t = 0.7 * x_{t-1} + w_t$ en exécutant le code suivant.

```
set.seed(154)
w = rnorm(550,0,1) # 50 extra to avoid startup problems
x = filter(w, filter=c(.7), method="recursive")[-(1:50)]
```

- (b) Tracer le graphe des autocorrélations et autocorrélation partielles associées en exécutant le code suivant. Commenter.

```
par(mfrow=c(2,1))
plot.ts(w, main="white noise")
plot.ts(x, main="autoregression")
par(mfrow=c(2,1))
acf(x, type="p", plot=T)
pacf(x)
```

- (c) Refaire la même chose en exécutant le code suivant. Commenter.

```
set.seed(154)
xarima=arima.sim(n=550, list(ar=c(0.7)), sd=1)[- (1:50)] # (AR)
x=xarima
par(mfrow=c(2,1))
plot.ts(x, main="autoregression")
plot.ts(xarima, main="autoregression")
```

- (d) Refaire la même chose avec cette fois-ci $x_t = 0.7 * x_{t-1} + 5 * w_t$. Commenter.
 (e) Refaire la même chose cette fois-ci pour un AR(1) $x_t = 0.9999 * x_{t-1} + w_t$. Commenter
 (f) Refaire la même chose cette fois-ci pour un AR(1) $x_t = -0.9999 * x_{t-1} + w_t$. Commenter

- (g) Simuler un AR(1) $x_t = -2 * x_{t-1} + w_t$. Que peut-on dire de son autocorrélation et de son autocorrélation partielle. Les tracer et commenter.
- (h) Simuler un AR(2) $x_t = 0.5x_{t-1} - 0.25x_{t-2} + w_t$, puis tracer son autocorrélation et son autocorrélation partielle. Commenter.
- (i) Refaire la même chose avec un AR(2) : $x_t = 4x_{t-1} - 4x_{t-2} + w_t$. Commenter.
- 2. Simulation de processus MA.
 - (a) Simulation de processus MA(1) : $v_t = \theta * w_{t-1} + w_t$ pour différentes valeurs de θ . Puis tracer son autocorrélation et autocorrélation partielle. Commenter.
 - (b) Simulation de processus MA(2) : $v_{t+1} = 1/3 * w_t + (2/3)w_{t-1} + w_{t-2}$, puis tracer son autocorrélation et autocorrélation partielle. Commenter.
- 3. Simulation de marches aléatoires. Soit $w_t \sim \mathcal{N}(0, 1)$.
 - (a) Simulation d'une marche aléatoire de la forme $x_t = \sum_{j=1}^t w_j$. Puis tracer son autocorrélation et autocorrélation partielle. Commenter. Proposer et mettre en oeuvre un test de la stationnarité de ce processus.
 - (b) Simulation d'une marche aléatoire de la forme $x_t = \sum_{j=1}^t w_j + 0.2$. Puis tracer son autocorrélation et autocorrélation partielle. Commenter. Proposer et mettre en oeuvre un test de la stationnarité de ce processus.
 - (c) Refaire la même chose pour une marche aléatoire avec tendance déterministe.
- 4. Considérons un processus de la forme $x_t = 2 * \cos(2 * \pi * t / 50 + .6 * \pi) + \sigma w_t$ où w_t est un bruit blanc gaussien standard. Le simuler en faisant varier σ . Puis tracer son autocorrélation et autocorrélation partielle. Commenter. Proposer et mettre en oeuvre un test de la stationnarité de ce processus.
- 5. Simulation de processus ARMA (p,q) en exécutant le code suivant. Quel est l'ordre de ce modèle simulé ?
- 6. Mettre en oeuvre une méthode d'identification de l'ordre d'un ARMA en utilisant la fonction `arma.scan` à l'adresse <http://www.math-evry.cnrs.fr/members/mtaupin/welcome>.

```
#####Simulation d'un ARMA
x=arima.sim(n=500, list(ar=c(0.5), ma=c(0.9)), sd=1) #ARMA
plot(x)
```

Exercice 2

On se propose d'étudier le jeu de données *AirPassengers* du package *tseries* de *R*.

1. Trouver le descriptif du jeu de données dans *R*.
2. Tracer la série temporelle associée.
3. Proposer une transformation de la série de façon à la rendre homoscédastique.
4. On travaille maintenant sur la série transformée.
5. Etudier la tendance "déterministe" de la série.
6. Saisonnalité ? prédiction de holt-winters
7. Proposer une transformation de la série qui la rende stationnaire. A l'aide d'un test adapté, vérifier que la série obtenue est stationnaire.
8. Tracer les fonctions d'autocorrélation et d'autocorrélation partielle e la série stationnaire. Commenter.
9. Proposer plusieurs modèles ARMA possibles. Estimer leurs coefficients.
10. Pour chacun de ces modèles, étudier les résidus.

```
##### TP Analyse d'un jeu de données réelles #####
#Lecture des données
help(AirPassengers)
data=log(AirPassengers)
data=ts(data,frequency=12,start=c(1949,1))
plot(AirPassengers)
plot(data, main="log du nbre mensuel de passagers", xlab="année",
      ylab="")

#Représentation de la tendance
dataMOVA=filter(data,filter=rep(1/12,12), sides=2)
dataMOVA=filter(data,filter=rep(1/12, 12))
dataMOVA=filter(data,filter=c(1/24,rep(1/12,11),1/24))
plot(data, main="log du nbre mensuel de passagers (en millier)",
      xlab="année",
      ylab="")
lines(dataMOVA,col="red")
y=as.vector(data)
x=as.vector(time(data))
reg=lm(y~poly(x,3))
lines(predict(reg)~x)

#Prévision Holt-Winters
data.hw<-HoltWinters(data)
predict(data.hw, n.ahead=12)
plot(data, main="log du nbre mensuel de passagers (en millier)",
      xlab="année",
      ylab="", xlim=c(1949,1962),ylim=c(4.6, 6.5))
lines(predict(data.hw, n.ahead=12), col=2)

#Stationnarisation
y=diff(data,lag=12,differences=1)
x=diff(y,lag=1,differences=1)
plot(y)
plot(x)
mean(x)
adf.test(data, alternative=c("stationary"),12)
adf.test(y, alternative=c("stationary"),12)
adf.test(x, alternative=c("stationary"),12)

#Autocorrélations
acf(x)
pacf(x)

#modèle ARIMA
z<-arima(data, order=c(3,1,1), seasonal=list(order=c(1,1,1), period=12))
z<-arima(data, order=c(3,1,1), seasonal=list(order=c(0,1,1), period=12))
z<-arima(data, order=c(2,1,1), seasonal=list(order=c(1,1,1), period=12))
z<-arima(data, order=c(2,1,1), seasonal=list(order=c(0,1,1), period=12))
z<-arima(data, order=c(1,1,1), seasonal=list(order=c(1,1,1), period=12))
z<-arima(data, order=c(1,1,1), seasonal=list(order=c(0,1,1), period=12))
z<-arima(data, order=c(0,1,1), seasonal=list(order=c(0,1,1), period=12))
```

```
#Etude des résidus
tsdiag(z, gof.lag=20)
plot(z$residuals)
shapiro.test(z$residuals)
hist(z$residuals,main="Histogramme
      des r\ 'esidus",
      freq=F)
curve(dnorm(x,mean(z$residuals),sqrt(var(z$residuals))),
      add=TRUE)

#Prévision ARIMA
prev<-predict(z, n.ahead=12)
plot(data, main="log du nbre mensuel de passagers (en millier)", xlab="année",
      ylab="", xlim=c(1949,1962),ylim=c(4.6, 6.5))
lines(prev$pred,col="blue")
lines(predict(data.hw, n.ahead=12), col=2)
tl <- prev$pred - 1.96 * prev$se
tu <- prev$pred + 1.96 * prev$se
ts.plot(data, tl, tu , lty=c(1,2,2),
        main="log du nbre mensuel de passagers (en millier) et
prévision sur 1 an", xlab="année")
lines(prev$pred,col="blue")
ts.plot(AirPassengers, exp(tl), exp(tu), lty=c(1,2,2),
        main="nombre mensuel de passagers (en millier)
et prévision sur 1 an", xlab="année")
lines(exp(prev$pred), col="blue")
```