Projet: Séries Temporelles

UNIVERSITÉ DE PARIS / MASTER 1 INGÉNIERIE MATHÉMATIQUE ET BIOSTATISTIQUE

James Kelson LOUIS

4/13/2020

Chargement des données

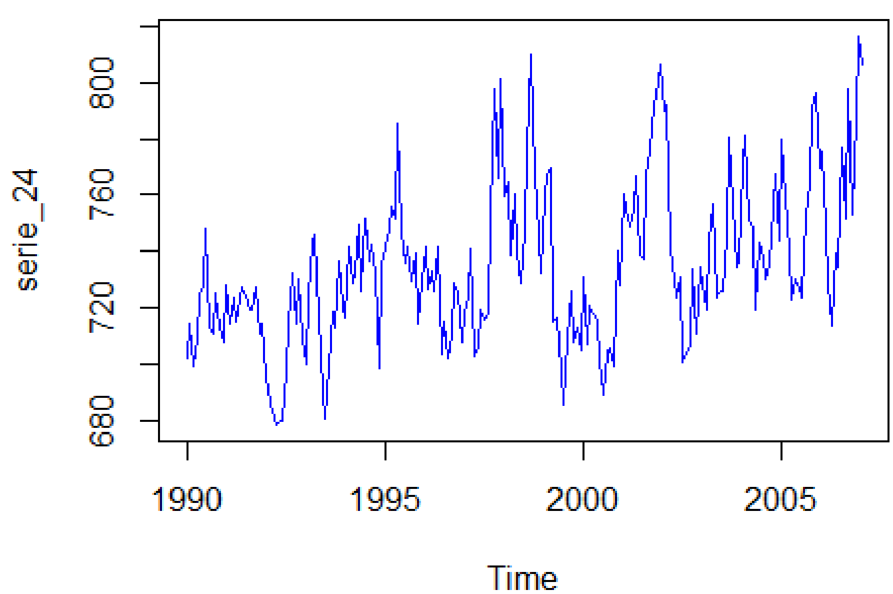
```
serie_24 <-
read.table("C:/Users/james/OneDrive/Desktop/DocParisDescartes/S2/SeriesTemporelles/Projet_final/Fichiers/serie_24.dat", quote="\"", comment.char="")
```

Transformation en série temporelle

```
serie_24 <- ts(serie_24,frequency = 12,start=c(1990,1))
```

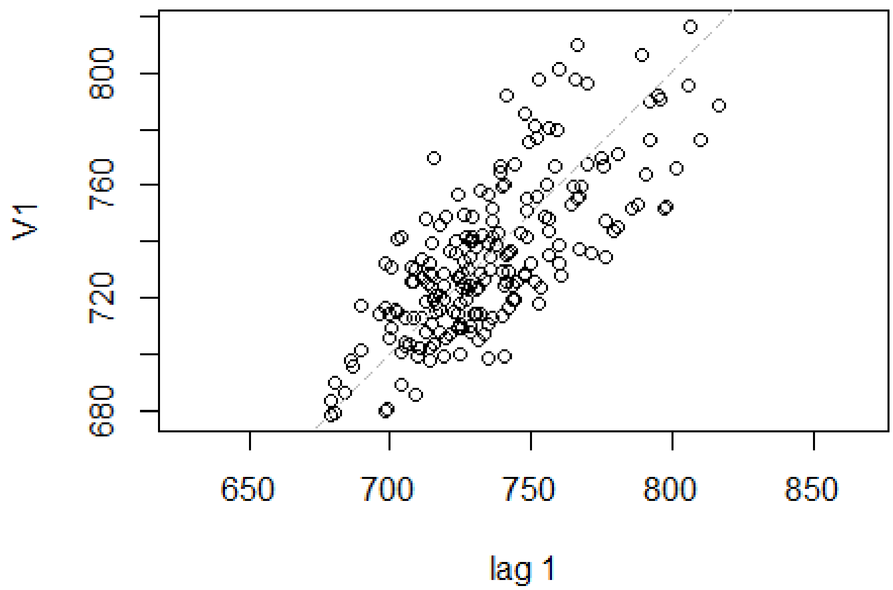
Représentation de la série

```
ts.plot(serie_24, col="blue")
```



Affichons le lag.plot de la série

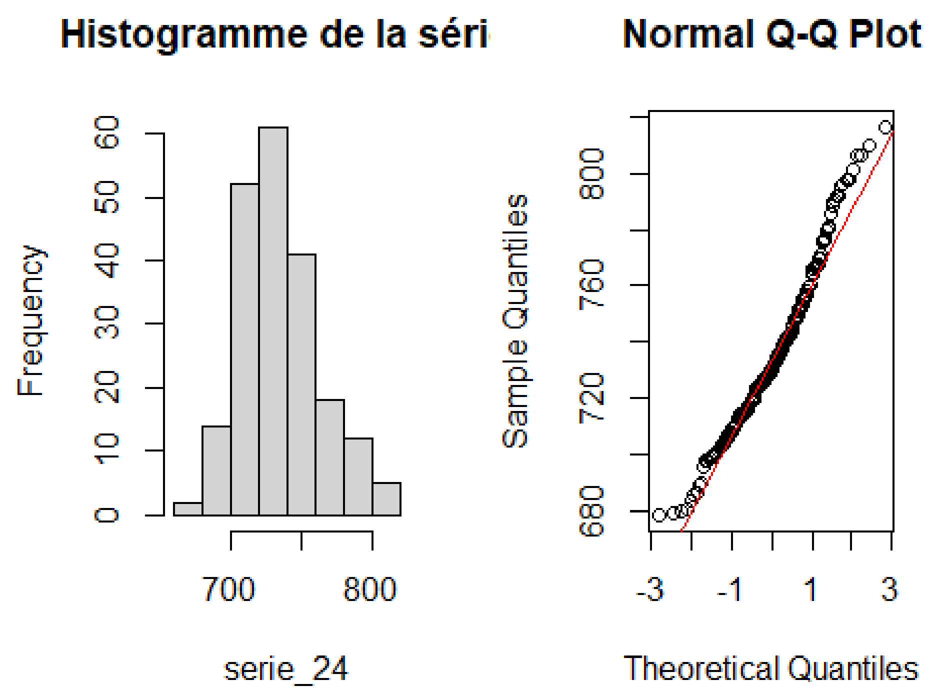
```
lag.plot(serie_24)
```



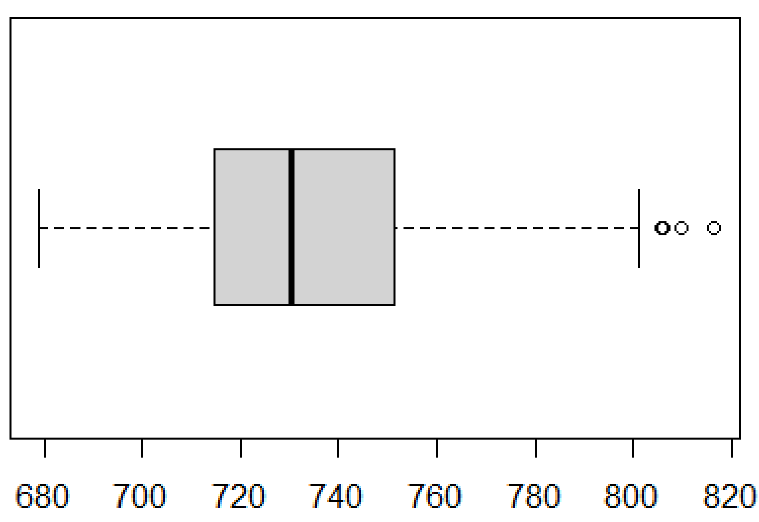
On constate une relation linéaire, une tendance linéaire positive qui suggère une autocorrélation positive.

Affichons l'histogramme et le Q-Q Plot de la série

```
par(mfrow=c(1,2))
hist(serie_24, main="Histogramme de la série")
qqnorm(serie_24)
qqline(serie_24, col='red')
```



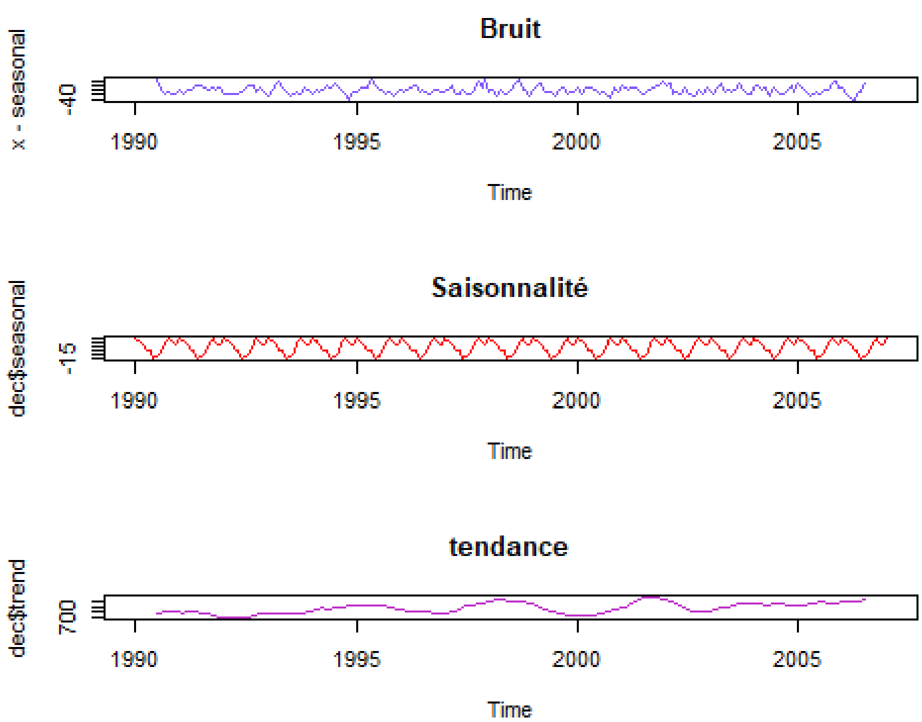

```
boxplot(serie_24, horizontal=T)
```



En moyenne toutes les valeurs se trouvent entre 700 et 760, sauf quelques unes qui dépassent 800.

Représentation du bruit, de la saisonnalité et de la tendance.

```
dec <- decompose(serie_24)
par(mfrow=c(3,1))
plot(dec$random,type="l",main="Bruit",col="#8258FA");plot(dec$seasonal,type="
l",main="Saisonnalité",col="red");plot(dec$trend,type="l",main="tendance",col
="#bd2ec6")
```



Faisons la décomposition en tendance, saisonnalité et bruit de la série.

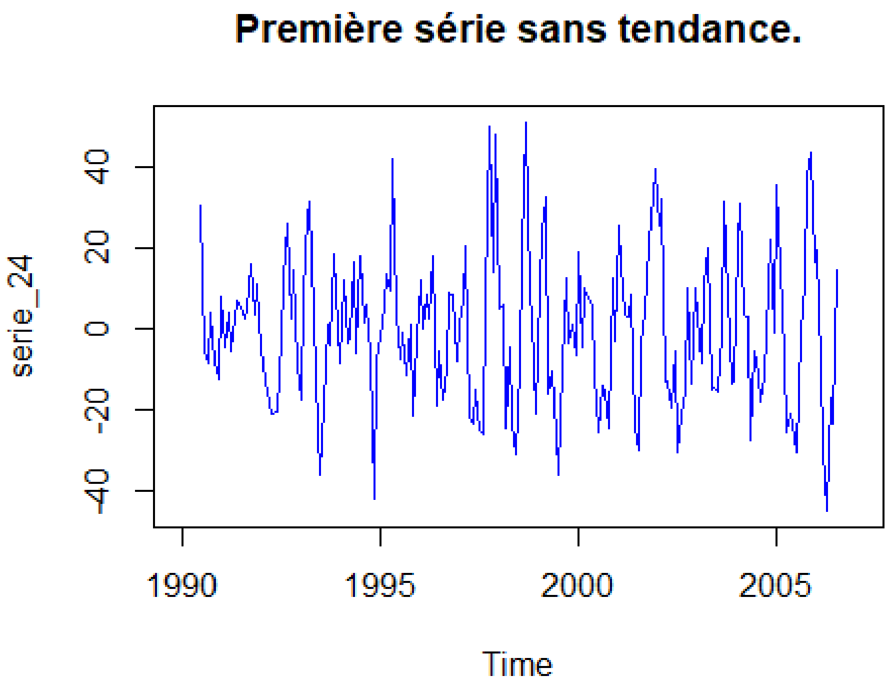
Application d'une moyenne mobile.

Première élimination de la tendance.

```
n <- length(serie_24)
p=12
m=p/2
X_star=rep(NA, n)
for (i in (m+1):(n-m)) {
X_star[i] <- 1/12*(1/2*serie_24[i-m]+
sum(serie_24[(i-m+1):(i+m-1)])+1/2*serie_24[i+m])
}
X_star <- ts(X_star,start=c(1990,1),frequency = 12)
```

Première série sans tendance.

```
X_moinsX_star <-serie_24-X_star
plot(X_moinsX_star, type="l",main="Première série sans tendance.",col="blue")
```



Premier calcul de saisonnalité.

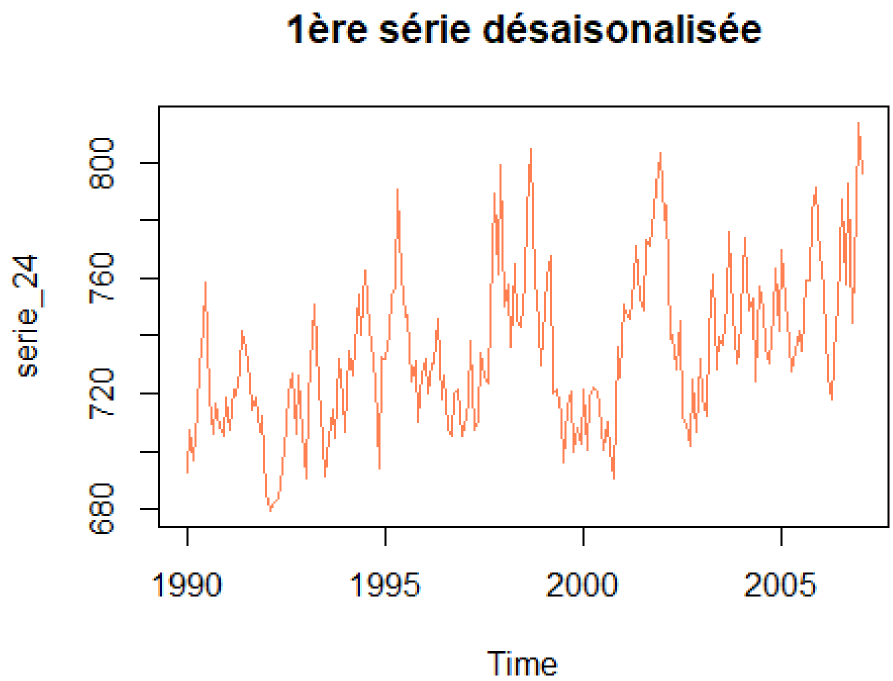
```
X_moinsX_star<- c(X_moinsX_star,rep(0,11))
mat <- matrix(X_moinsX_star,ncol =p, byrow = T)
moy_col <- apply(mat, 2, mean,na.rm=T)
```

Estimation

```
s_chapo <- numeric(12)
for (i in 1:12) {
s_chapo[i] <- moy_col[i]-mean(moy_col)
}
B <- rep(s_chapo,18)
# On supprime les elements pour avoir 205 valeurs
C <- B[1:(length(B)-11)]
ts_s_chapo<- ts(C, start=c(1990,1),frequency = 12)
```

Première série désaisonnalisée

```
Xcvs<- serie_24-ts_s_chapo
plot(Xcvs,type="l",col="#FF7F50",main="1ère série désaisonalisée")
```

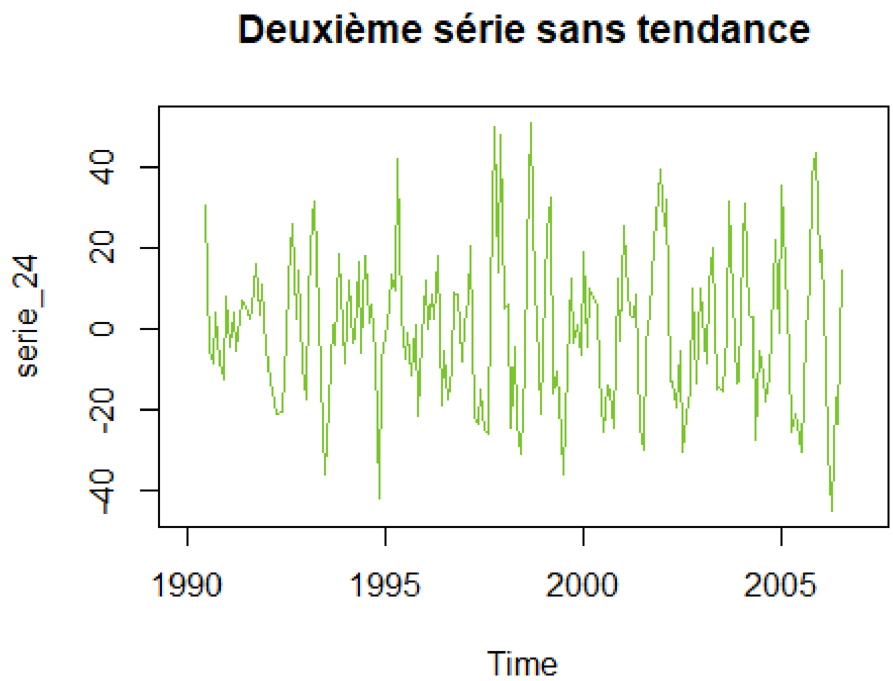


Deuxième calcul d’élimination de tendance.

```
X_star2=rep(NA, n)
for (i in (m+1):(n-m)) {
X_star2[i] <- 1/12*(1/2*Xcvs[i-m]+sum(Xcvs[(i-m+1):(i+m-1)])+1/2*Xcvs[i+m])
}
X_star2 <- ts(X_star2,start = c(1990,1),frequency = 12)
```

Deuxième série sans tendance

```
X_moinsX_star2 <- serie_24-X_star2
plot(X_moinsX_star2,type="l",main="Deuxième série sans
tendance",col="#7ac62e")
```



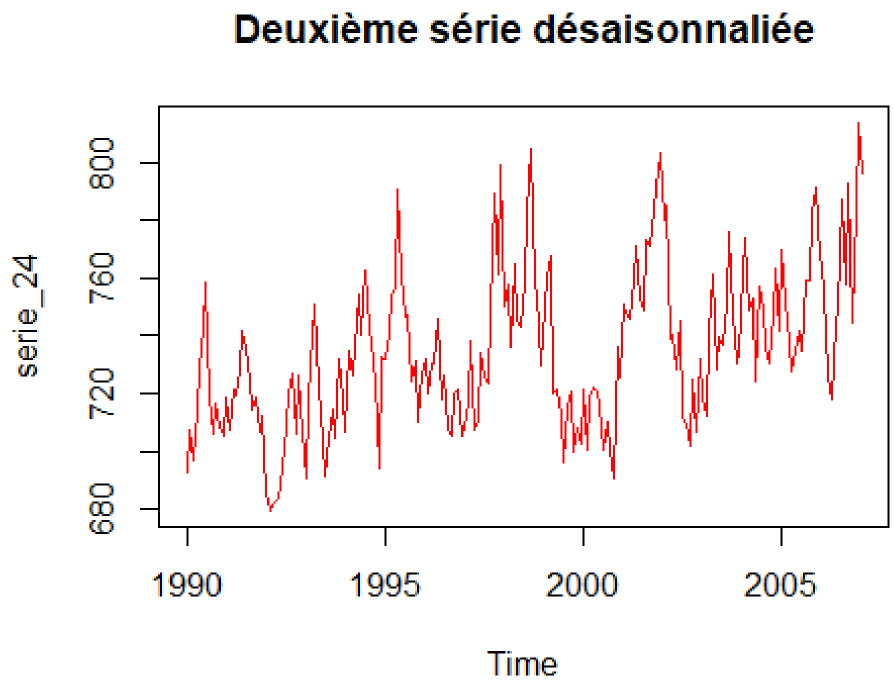
Deuxième calcul de saisonnalité.

```
X_moinsX_star3 <- c(X_moinsX_star2,rep(0,11))
mat2 <- matrix(X_moinsX_star3,ncol =p, byrow = T)
moy_col2 <- apply(mat2, 2, mean,na.rm=T)
moy_col2

## [1]   9.694011   7.017065   2.612485  -4.388547  -4.672562 -14.262959
## [7] -10.858021  -5.914718   4.921748   8.326628   4.234836   2.500091
```

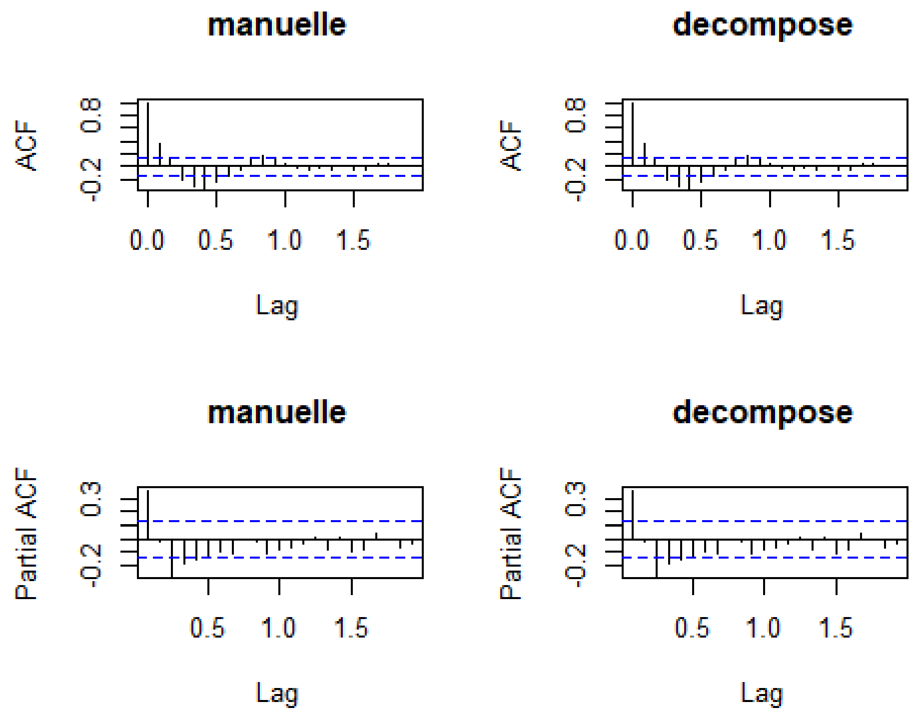
Estimation de la saisonnalité.

```
s_chapo2 <- numeric(12)
for (i in 1:12) {
s_chapo2[i] <- moy_col2[i]-1/12*sum(moy_col2)
}
D <- rep(s_chapo2,18)
E <- D[1:(length(B)-11)]
ts_s_chapo2<- ts(E, start=c(1990,1),frequency = 12)
plot(serie_24- ts_s_chapo2,type="l",main="Deuxième série
désaisonnaliée",col="red")
```



Les tendances et saisonnalités sont stables. Vérifions si l'acf et la pacf correspondent aux acf et pacf obtenus avec la fonction decompose de R.

```
par(mfrow=c(2,2))
res=serie_24-X_star2-ts_s_chapo2
acf(res,na.action = na.pass,main="manuelle")
acf(dec$random, na.action = na.pass, main="decompose")
pacf(res,na.action = na.pass, main="manuelle")
pacf(dec$random, na.action = na.pass, main="decompose")
```



On constate que les acfs et pacfs réalisés à la main sont identiques avec ceux réalisés avec la fonction decompose de R. Au vu de l'acf il n'est pas illégitime de considérer que ses coefficients sont nuls à partir de h=7, dans ce cas on peut proposer un modèle MA(6). Pour la pacf, on peut accepter que ces coefficients sont nuls à partir de h=6, donc il est tout à fait louable de proposer un AR(5).

En appliquant la moyenne mobile sur la série, on obtient 6 valeurs manquantes au début et à la fin, pour éviter des problèmes lors du calcul des erreurs de prédictions, on supprime les valeurs manquantes dans les résidus.

```
res1 <- window(res, start=c(1990,7),end=c(2006,7), frequency=12)
length(res1)

## [1] 193
```

On découpe la série (res), et on garde les cinqs dernières valeurs qui serviront à comparer la performance des modèles proposés.

```
new_serie=window(res1, start=c(1990,7),end=c(2006,2),frequency=12)
vrai=window(res, start=c(2006,3),c(2006,7),frequency=12)
```

Proposition de modèles et estimation des coefficients

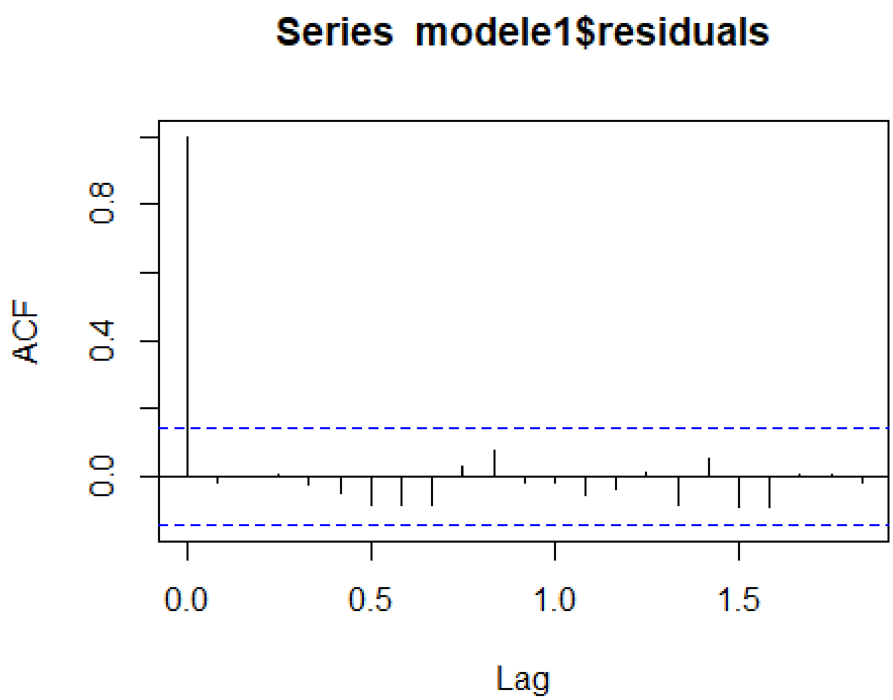
On commence par proposer un MA(6) puis on essaiera de simplifier afin de diminuer le nombre de paramètres à estimer.

```
(modele1=arima(new_serie,order=c(0,0,6)))
```

```
##
## Call:
## arima(x = new_serie, order = c(0, 0, 6))
##
## Coefficients:
##          ma1      ma2      ma3      ma4      ma5      ma6  intercept
##       0.1743  -0.0342  -0.3503  -0.3562  -0.2860  -0.1476     0.0174
## s.e.  0.0754   0.0716   0.0665   0.0741   0.0689   0.0729     0.0827
##
## sigma^2 estimated as 185.3:  log likelihood = -759.63,  aic = 1535.26
```

Testons la blancheur des résidus

```
acf(modele1$residuals, na.action = na.pass)
```



Tous les coefficients sont à l'intérieur de l'intervalle de confiance, donc on accepte la blancheur des résidus.

Faisons une prédiction puis calculons l'erreur de prédiction.

```
prev1=predict(modele1,n.ahead=5,ci=TRUE)
sqrt(mean((prev1$pred-vrai)^2))

## [1] 18.78155
```

A l'aide de des intervalles de confiance, nous allons tester la significativité des coefficients.

```
confint(modele1)
```

```
##                 2.5 %       97.5 %
## ma1        0.02647165  0.322084471
## ma2       -0.17445925  0.106113378
## ma3       -0.48067014 -0.219831574
## ma4       -0.50145873 -0.210969604
## ma5       -0.42103138 -0.150954412
## ma6       -0.29047267 -0.004812676
## intercept -0.14478731  0.179538178
```

0 appartient aux intervalles de confiances de ma1 et ma2, par contre on ne peut pas enlever ces paramètres, car les autres sont significatifs.

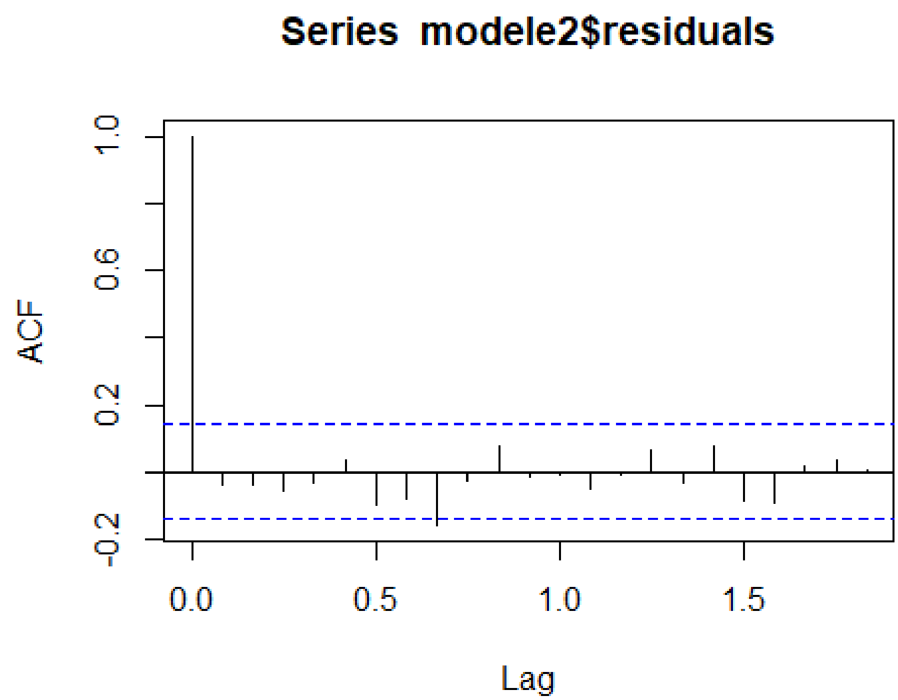
Proposition d'un deuxième modèle

```
(modele2=arima(new_serie,order=c(5,0,0)))

## 
## Call:
## arima(x = new_serie, order = c(5, 0, 0))
## 
## Coefficients:
##          ar1     ar2      ar3      ar4      ar5  intercept
##       0.2982  0.0596  -0.1883  -0.1243  -0.1451     0.1553
## s.e.  0.0734  0.0763   0.0750   0.0766   0.0744     0.9643
## 
## sigma^2 estimated as 208.5:  log likelihood = -769.01,  aic = 1552.01
```

Testons la blancheur des résidus

```
acf(modele2$residuals,na.action = na.pass)
```



On accepte la blancheur des résidus

Prévision2

```
prev2=predict(modele2,n.ahead=5,ci=TRUE)
sqrt(mean((prev2$pred-vrai)^2))

## [1] 16.98964
```

Testons la significativité des coefficients

```
confint(modele2)

##                  2.5 %        97.5 %
## ar1        0.15430388  0.4420054486
## ar2       -0.08988203  0.2091795780
## ar3       -0.33534855 -0.0412545168
## ar4       -0.27448609  0.0259122989
## ar5       -0.29085435  0.0006671688
## intercept -1.73470001  2.0452577804
```

On enlève le coefficient ar5

Proposition d'un troisième modèle

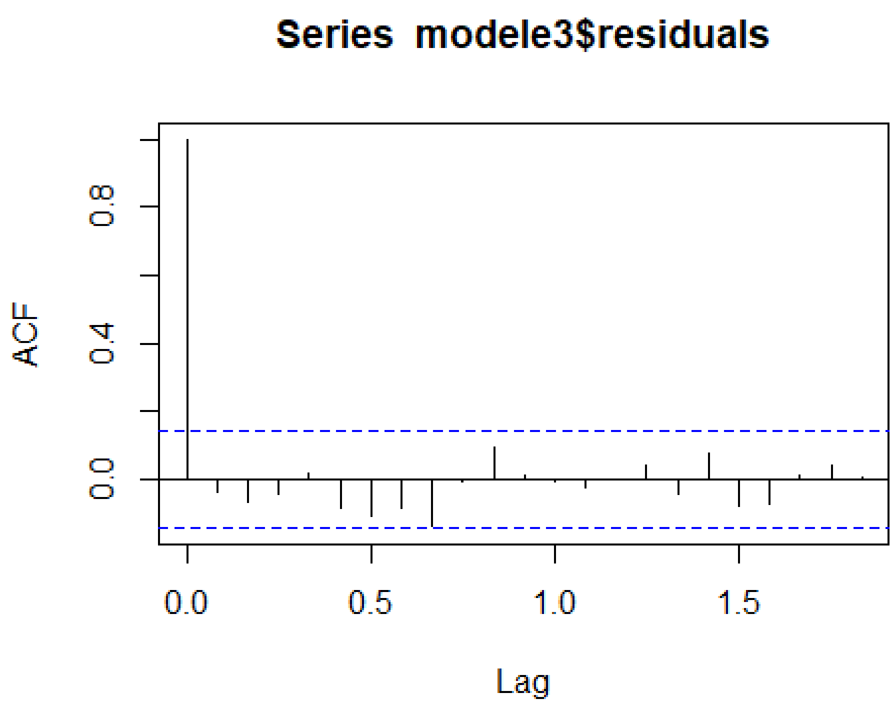
```
(modele3=arima(new_serie,order=c(4,0,0)))

##
## Call:
## arima(x = new_serie, order = c(4, 0, 0))
```

```
##
## Coefficients:
##          ar1     ar2      ar3      ar4  intercept
##       0.3207  0.0884  -0.2022  -0.1678     0.2028
## s.e.  0.0732  0.0757   0.0755   0.0741     1.1124
##
## sigma^2 estimated as 212.8:  log likelihood = -770.89,  aic = 1553.78
```

Testons la blancheur des résidus

```
acf(modele3$residuals, na.action=na.pass)
```



On accepte la blancheur des résidus

Prévisions et calcul de l’erreur de prévisions

```
prev3=predict(modele3,n.ahead=5,ci=TRUE)
sqrt(mean((prev3$pred-vrai)^2))

## [1] 19.9352
```

Testons la significativité des coefficients

```
confint(modele3)

##                2.5 %      97.5 %
## ar1        0.1771425  0.46424905
## ar2       -0.0598837  0.23671654
## ar3       -0.3501745 -0.05422414
```

```
## ar4       -0.3129874 -0.02266357
## intercept -1.9775129  2.38310178
```

On ne peut plus simplifier car le coefficient ar4 est significatif.

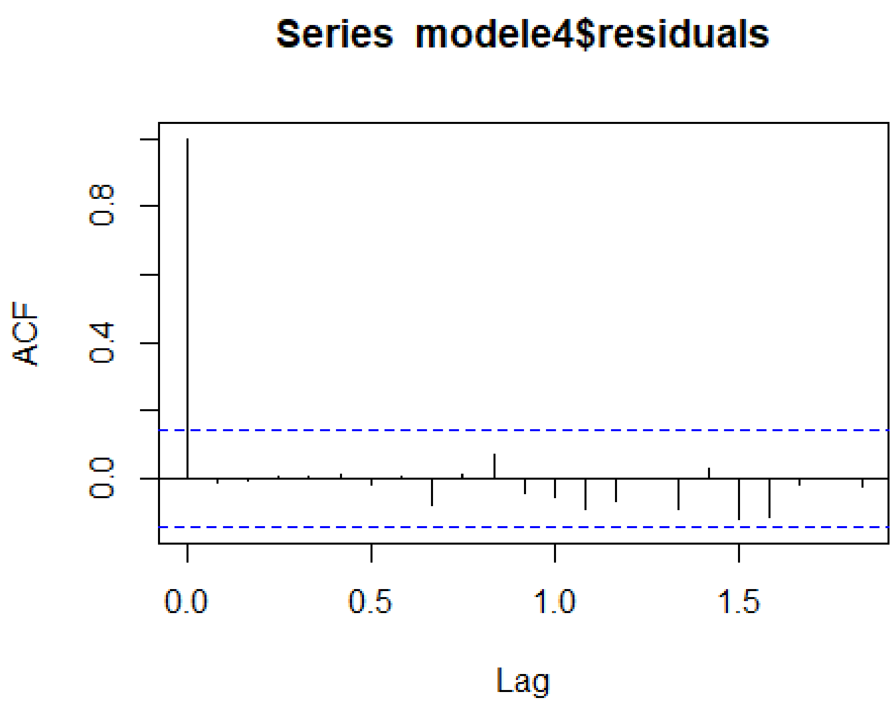
Essayons de proposer un ARMA(3,3)

```
(modele4=arima(new_serie,order=c(3,0,3)))

##
## Call:
## arima(x = new_serie, order = c(3, 0, 3))
##
## Coefficients:
##          ar1      ar2      ar3      ma1      ma2      ma3  intercept
##        0.967  -0.0152  -0.2923  -0.8194  -0.1717  -0.0089     0.0086
## s.e.  0.418   0.6290   0.3063   0.4232   0.5778   0.2041     0.0624
##
## sigma^2 estimated as 180.9:  log likelihood = -757.68,  aic = 1531.37
```

Testons la blancheur des résidus

```
acf(modele4$residuals, na.action=na.pass)
```



On accepte la blancheur des résidus

Prévisions et calcul de l'erreur de prévisions

```
prev4=predict(modele4,n.ahead=5,ci=TRUE)
sqrt(mean((prev4$pred-vrai)^2))
```

```
## [1] 17.38115
```

Test de significativité des coefficients

```r
confint(modele4)
```

```
##                2.5 %      97.5 %
## ar1        0.1476517 1.786293118
## ar2       -1.2480131 1.217531831
## ar3       -0.8925829 0.308054972
## ma1       -1.6488019 0.009981256
## ma2       -1.3040580 0.960728946
## ma3       -0.4089443 0.391157184
## intercept -0.1137549 0.130967618
```

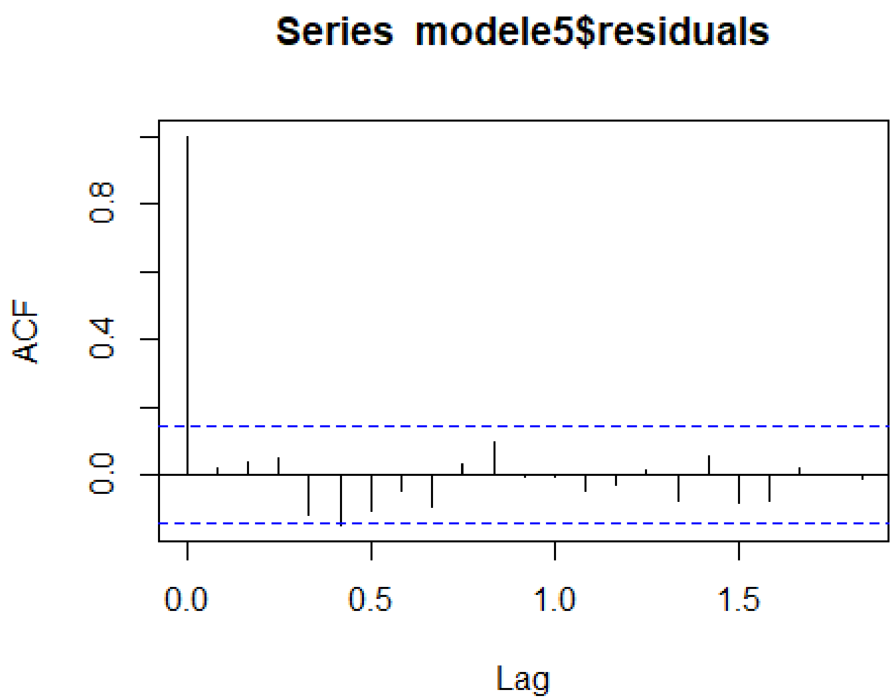
Modèle sans ar2 et ar3

```r
(modele5=arima(new_serie,order=c(1,0,3)))
```

```
##
## Call:
## arima(x = new_serie, order = c(1, 0, 3))
##
## Coefficients:
##          ar1      ma1      ma2      ma3  intercept
##       0.6453  -0.4886  -0.1658  -0.3456     0.0231
## s.e.  0.0727   0.0875   0.0857   0.0717     0.0933
##
## sigma^2 estimated as 192.2:  log likelihood = -762.9,  aic = 1537.81
```

Testons la blancheur des résidus

```r
acf(modele5$residuals, na.action=na.pass)
```



On accepte la blancheur des résidus

Prévisions et calcul de l'erreur de prévisions

```
prev5=predict(modele5,n.ahead=5,ci=TRUE)
sqrt(mean((prev5$pred-vrai)^2))

## [1] 20.86807
```

Test de significativité des coefficients

```
confint(modele5)

##                  2.5 %       97.5 %
## ar1        0.5027579  0.787850405
## ma1       -0.6599851 -0.317159258
## ma2       -0.3338122  0.002202571
## ma3       -0.4861724 -0.205070612
## intercept -0.1597243  0.205864258
```

On ne peut plus simplifier car les coefficient ar1 et ma3 sont significatifs.

Comparaison entre les modèles

```
Modèle <- c("MA(6)","AR(5)","AR(4)","ARMA(3,3)","ARMA(1,3)")
AIC <- c(1535.26,1552.01,1553.78,1531.37,1537.81)
Erreur <- c(18.78,16.98,19.93,17.38,20.86)
df=data.frame(Modèle=Modèle,AIC=AIC,Erreur=Erreur)
row.names(df) <- paste("MOD",1:5, sep="")
df
```

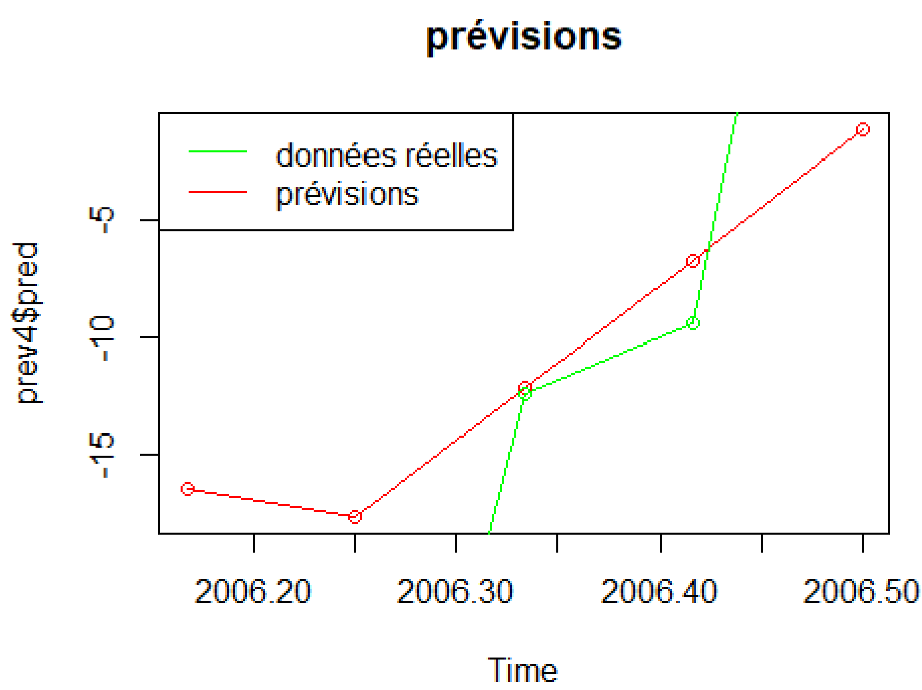
```
##          Modèle     AIC  Erreur
## MOD1      MA(6) 1535.26   18.78
## MOD2      AR(5) 1552.01   16.98
## MOD3      AR(4) 1553.78   19.93
## MOD4 ARMA(3,3) 1531.37   17.38
## MOD5 ARMA(1,3) 1537.81   20.86
```

Choix de modèle

En se basant sur le critère de l'AIC, le meilleur modèle est celui qui a le plus faible AIC, donc pour cette série il s'agit du ARMA(3,3).

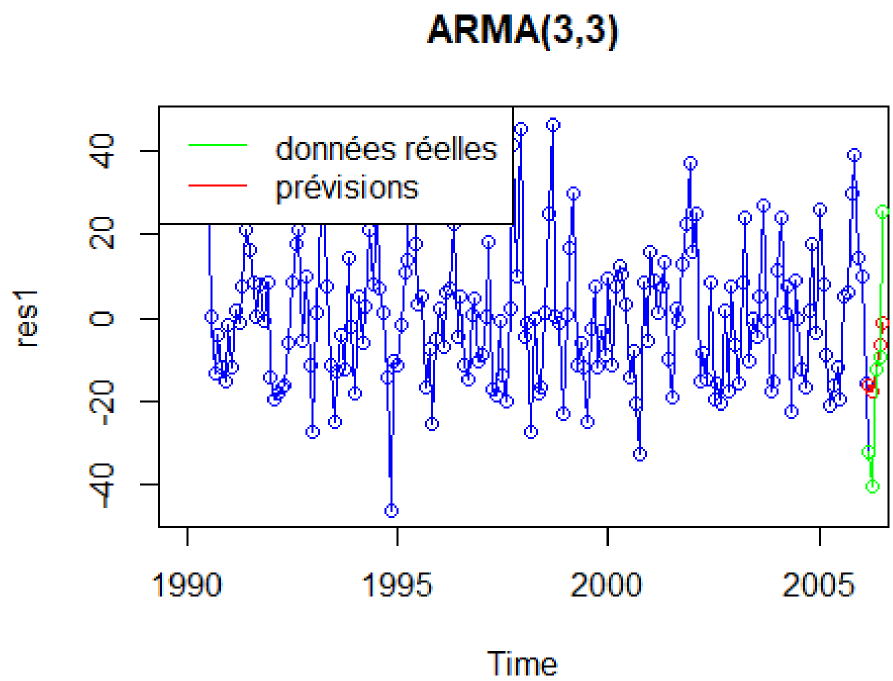
On affiche les valeurs prédites et les données réelles.

```
plot(prev4$pred, type="o",main="prévisions", col="red")
lines(vrai, type="o", col="green")
legend(x='topleft',legend=c('données
réelles','prévisions'),col=c('green','red'),lty=1)
```



Affichons sur un même graphique les données réelles (res1) ainsi que les valeurs prédites.

```
plot.ts(res1,type='o',ylab="res1",xlim=c(1990,2006),main='ARMA(3,3)',
col='blue')
lines(prev4$pred,col='red',type='o')
lines(vrai,col='green',type='o')
legend(x='topleft',legend=c('données
réelles','prévisions'),col=c('green','red'),lty=1)
```



ARMA(3,3)
données réelles
prévisions
res1
40
20
0
-20
-40
1990
1995
2000
2005
Time