Séries Temporelles 1 (2A) - ENSAI

TP 2 : Méthode de Box-Jenkins

José Gregorio GOMEZ-GARCIA

Novembre 2023

```
library(tidyr)
library(ggplot2)
```

Exercice 1

L'objectif de cet exercice est d'utiliser la méthode de Box-Jenkins pour modéliser les deux séries, `serie1.dat` et `serie2.dat`, disponibles sur moodle. On importera les données de la façon suivante :

```
s1 <- scan("serie1.dat")
s1 <- ts(s1,frequency=1)
```

1.- Rappeler rapidement la méthode de Box-Jenkins

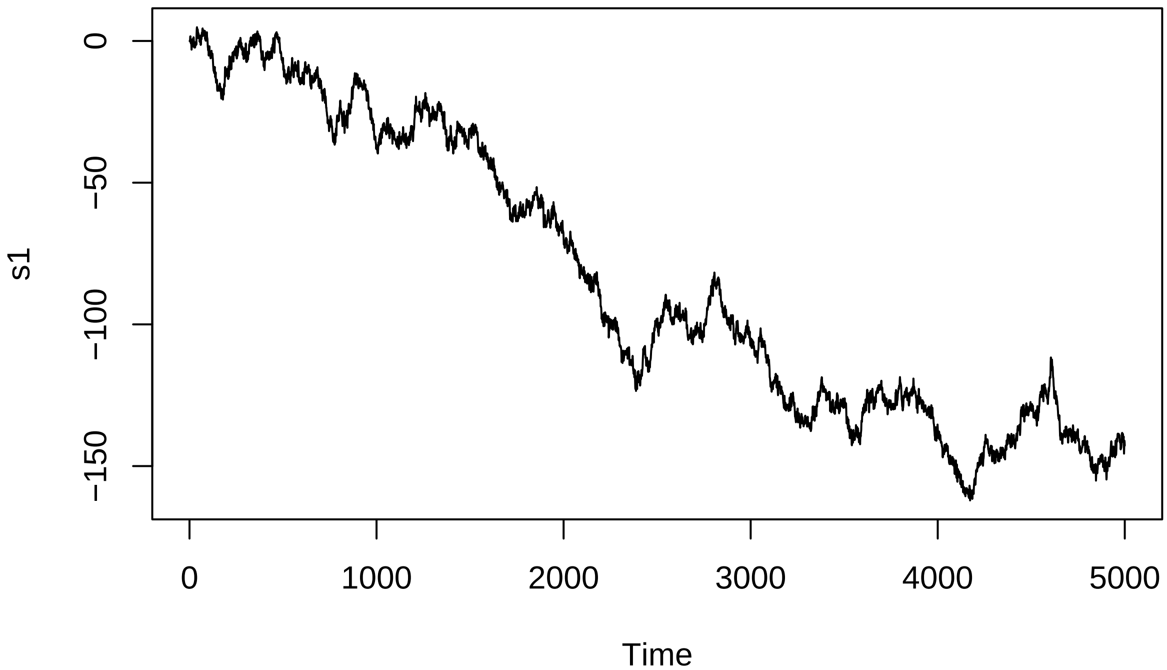
La démarche empirique (Box-Jenkins) pour effectuer une modélisation ARIMA(p,d,q) est la suivante :

- Stationnariser la série (régression, différenciation, transformation), vérifier avec l'autocorrélogramme (vous pouvez aussi faire un test de stationnarité).
- Déterminer des ordres p et q plausibles à l'aide de respectivement l'autocorrélogramme partiel et l'autocorrélogramme.
- Estimer les paramètres, si plusieurs modèles candidats les départager par l'AIC ou le BIC.
- Valider le modèle par un diagnostique des résidus (test, représentation graphique, autocorrélogramme).
- Confirmer votre choix en simulant de la prévision (échantillon test). *(Cette partie on ne la fera pas dans cet exercice)*

2.- Tracer l'allure de la série. La série est-elle stationnaire ? Si besoin, on utilisera la fonction `diff` pour la différencier. On déterminera les différents modèles possibles en utilisant l'`acf` et la `pacf`. On fera appel à la fonction `arima` pour chacun des modèles retenus. On utilisera `Box.test` pour vérifier la blancheur des résidus (`$residuals`) et on tracera leur `acf` et `pacf`. On comparera les différents modèles via leur AIC (`$aic`), pour choisir celui que l'on retiendra. Regardez aussi la log-vraisemblance (`$loglik`).

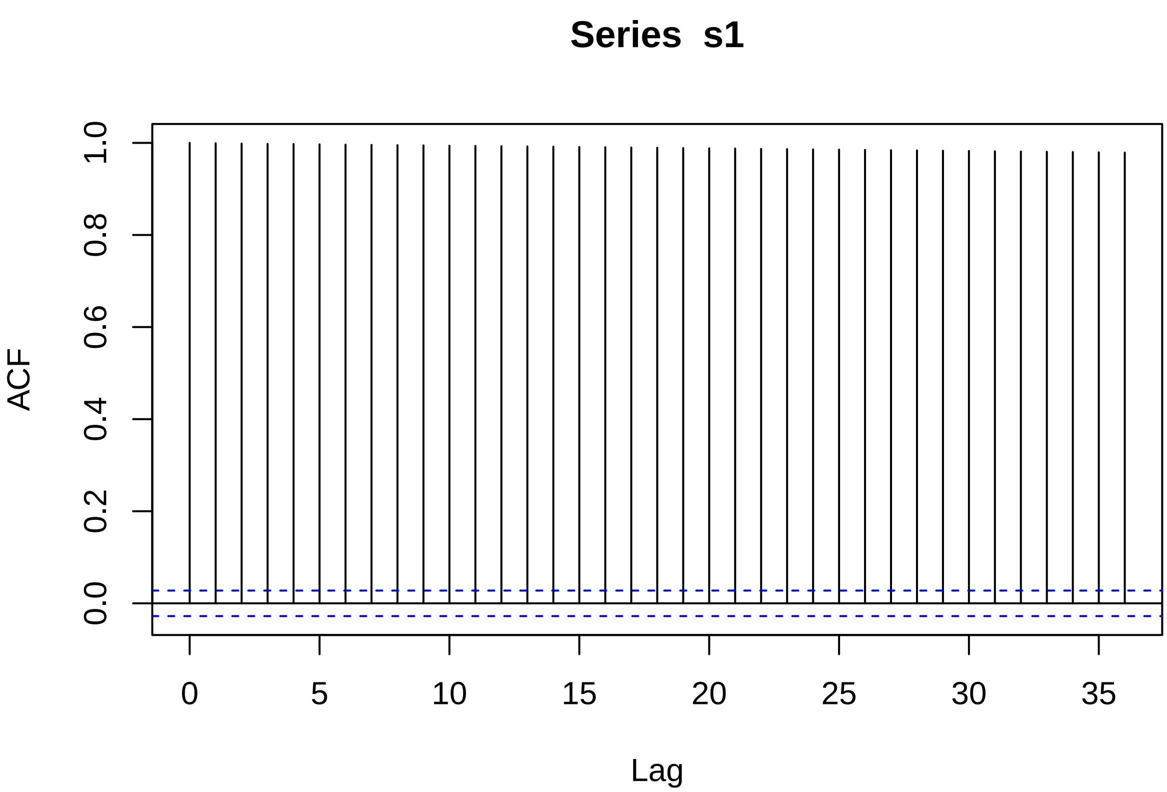
Allure de la série :

```
plot.ts(s1)
```



Il est clair que la série n'est pas stationnaire. On peut le vérifier en regardant l'`acf` : si la décroissance est lente (plus lente que l'exponentiel), on peut alors suspecter une non-stationarité.

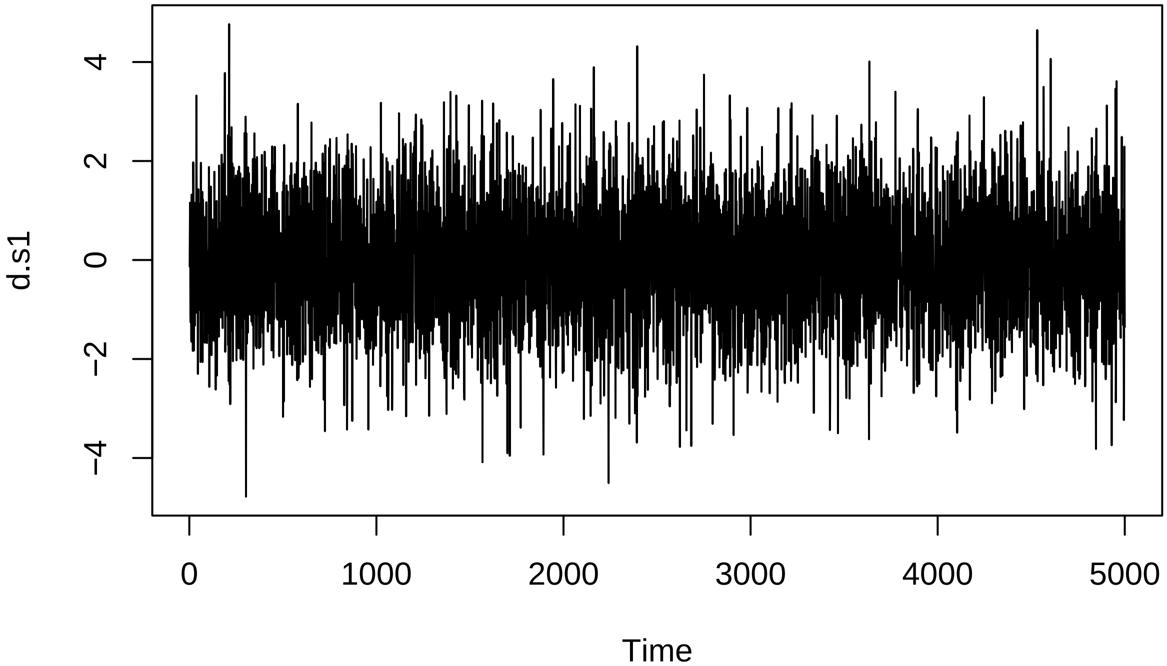
```
acf(s1)
```



Le résultat est clair ! On différence alors jusqu'à avoir une série stationnaire.

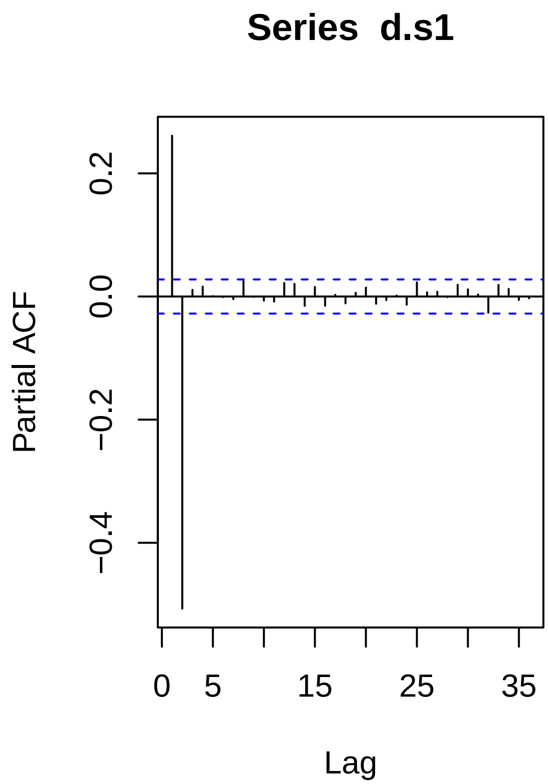
Attention : ce n'est pas recommandable de différencier plus de 2 fois ! En effet, dans la pratique le cas $d > 2$ est rarement rencotré et sur-différencier un processus peut conduire à une non-inversibilité des ARMA *associés.*

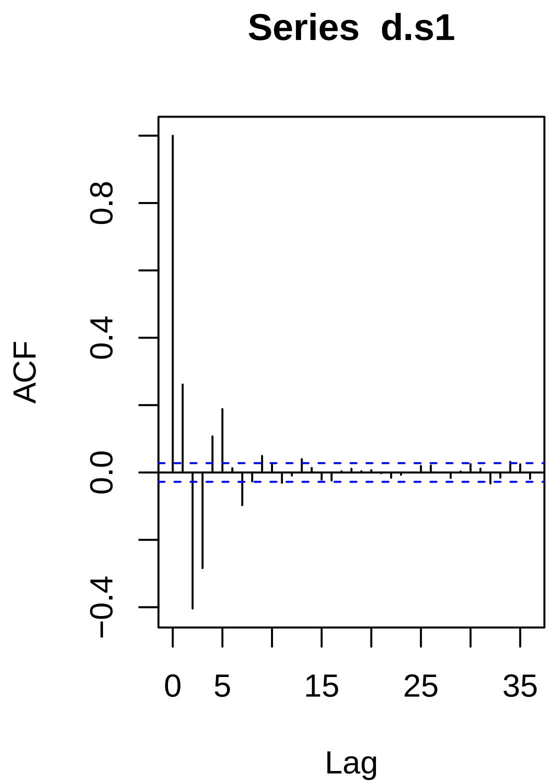
```
d.s1 <- diff(s1) #Ici, par defaut lag=1 et differences=1.
#C'est-a-dire diff(X_t) = (1-B)X_t.
plot.ts(d.s1)
```



La série differenciée semble stationnaire. On révérifie avec les `pacf` et `acf` :

```
par(mfrow=c(1,2))
pacf(d.s1)
acf(d.s1)
```





D'après les graphiques précédents, les ordres max possibles sont $p_{max} = 2$ et $q_m ax = 7$

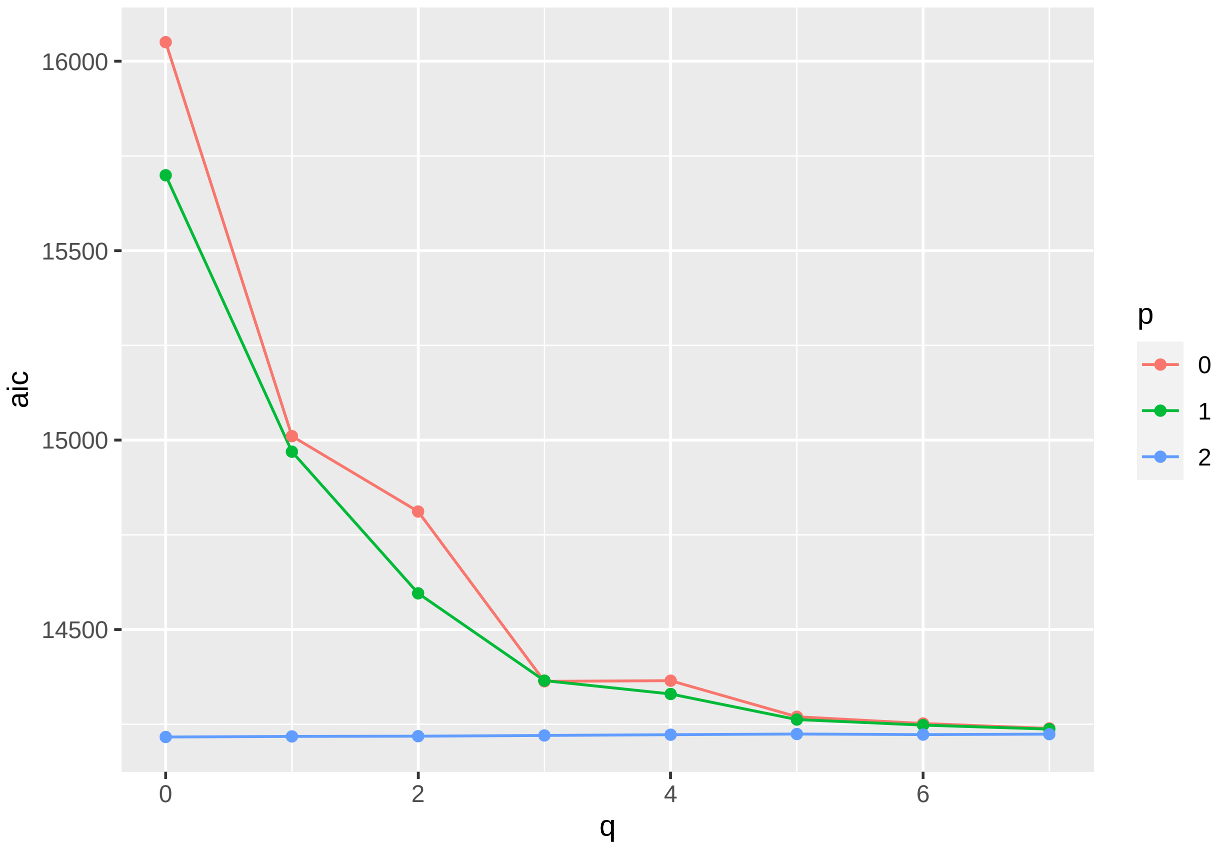
Maintenant, pour chaque modèle ajusté $ARMA(p, q)$, on calcule l'aic, la -log-vraisemblance et la p-valeur du Box-test sur les residus.

```
p.max = 2
q.max = 7
Results.aic <- matrix(NA, ncol = p.max+1, nrow = q.max+1)
Results.loglik <- matrix(NA, ncol = p.max+1, nrow = q.max+1)
Results.pvalue.res <- matrix(NA, ncol = p.max+1, nrow = q.max+1)
for(p in 0:p.max){
  for (q in 0:q.max){
    arma <- arima(d.s1, order = c(p,0,q))
    Results.aic[(q+1),(p+1)] <- arma$aic
    Results.loglik[(q+1),(p+1)] <- - arma$loglik
    Results.pvalue.res[(q+1),(p+1)] <- Box.test(arma$residuals)$p.value
  }
}
#
Results.aic.df <- data.frame(Results.aic)
colnames(Results.aic.df) <- c("0", "1", "2")
Results.aic.g <- gather(Results.aic.df, key = "p", value = "aic")
#
Results.loglik.df <- data.frame(Results.loglik)
colnames(Results.loglik.df) <- c("0", "1", "2")
Results.loglik.g <- gather(Results.loglik.df, key = "p", value = "loglik")
#
Results.pvalue.res.df <- data.frame(Results.pvalue.res)
colnames(Results.pvalue.res.df) <- c("0", "1", "2")
Results.pvalue.res.g <- gather(Results.pvalue.res.df, key = "p",
```

```
                                value = "p.value")
#
Results <- cbind(0:q.max, Results.aic.g, Results.loglik.g$loglik,
                 Results.pvalue.res.g$p.value)
colnames(Results) <- c("q", "p", "aic", "loglik", "p.value")
```

Voici les graphiques :

```
ggplot(Results, aes(x=q, y= aic, group=p)) +
  geom_line(aes(color=p))+
  geom_point(aes(color=p)) #+ ylim(14200, 14300)
```



Le graphe montre que $p = 2$. Pour être sur du bon q, on calcule :

```
which(Results$aic[Results$p==2] == min(Results$aic[Results$p==2]))
```

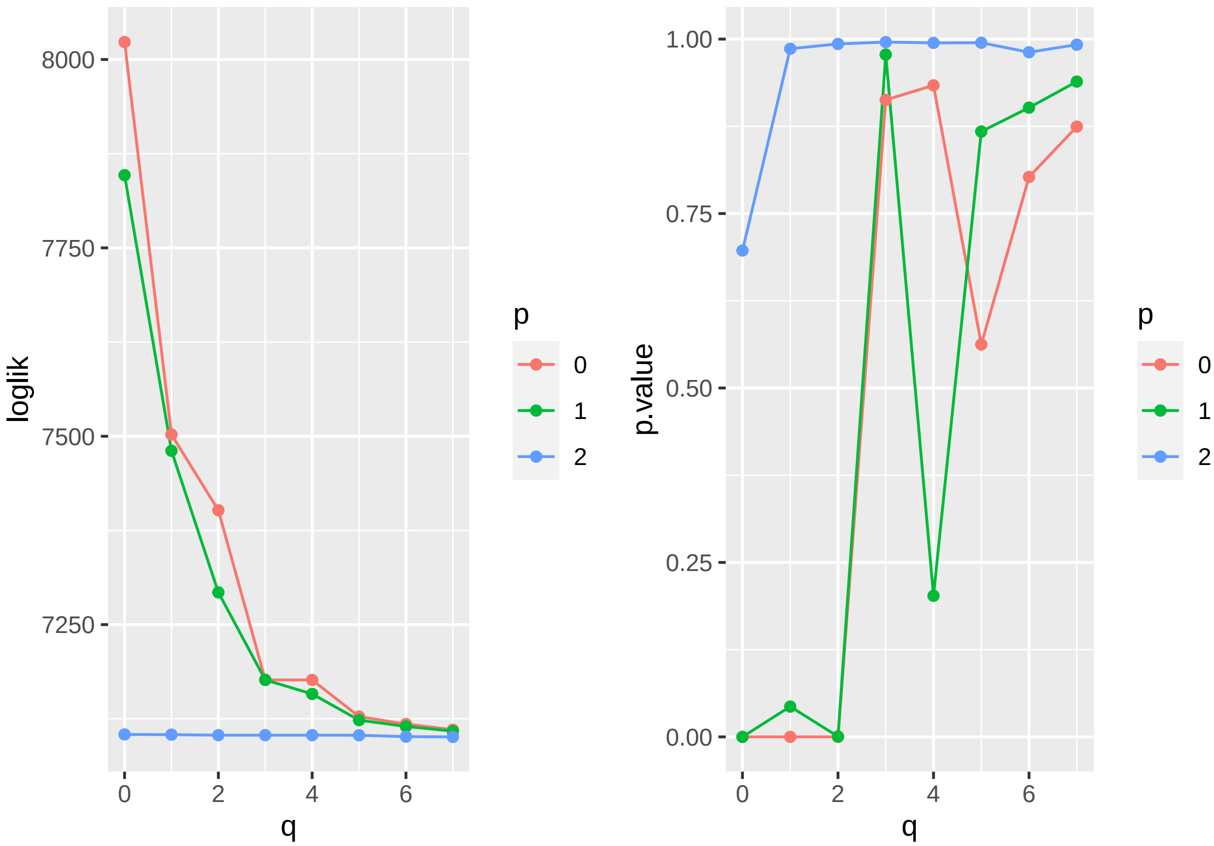
```
[1] 1
```

qui retourne 1. Cela veut dire que $q = 0$.

Pour le modèle choisi, (ici AR(2)), on peut vérifier si les residus sont gaussiens. Cependant, on va regarder cette information (ainsi que la -log-vraisemblance) pour chaque couple (p, q) par simple curiosité :

```
library(cowplot)
g1 <- ggplot(Results, aes(x=q, y= loglik, group=p)) +
  geom_line(aes(color=p))+
  geom_point(aes(color=p))# + ylim(7050, 7150)

g2 <- ggplot(Results, aes(x=q, y= p.value, group=p)) +
  geom_line(aes(color=p))+
```

```
  geom_point(aes(color=p))
plot_grid(g1,g2)
```



Cela ne contradit pas notre choix !

On regarde en détail le modèle retenu :

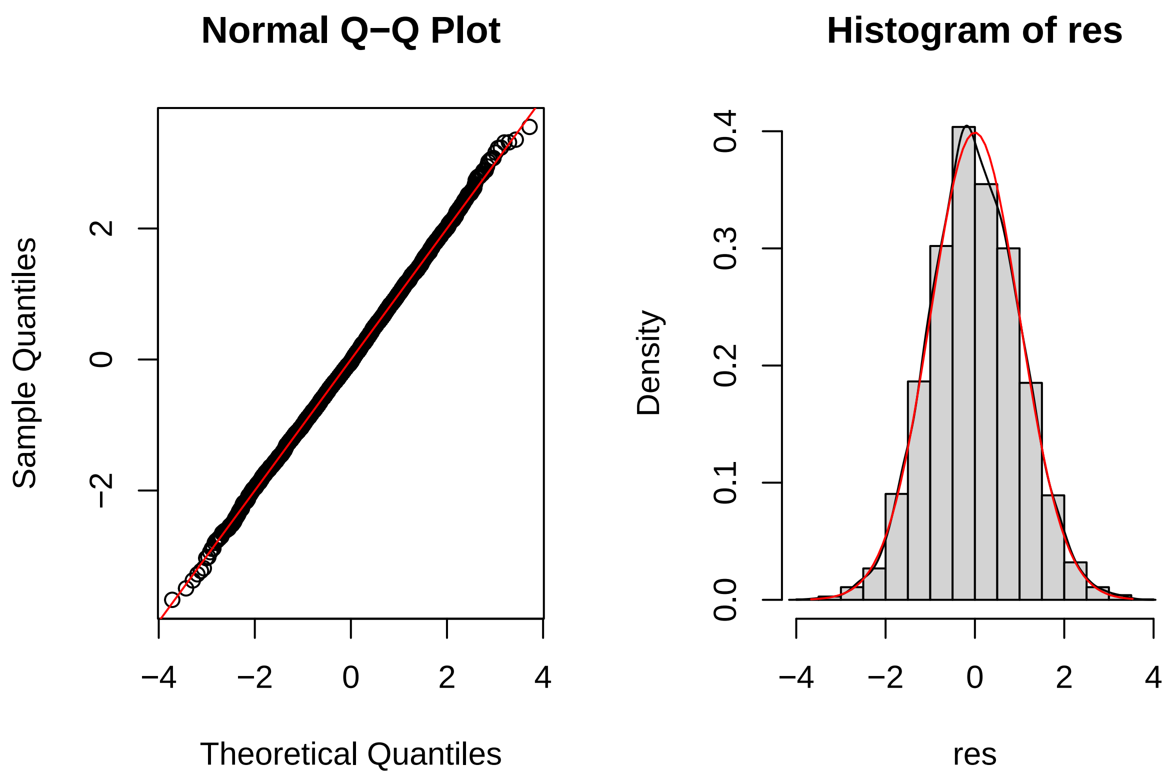
```
arma_final <- arima(d.s1, order = c(2, 0, 0))
arma_final
```

```

Call:
arima(x = d.s1, order = c(2, 0, 0))

Coefficients:
         ar1      ar2  intercept
      0.3936  -0.5068    -0.0286
s.e.  0.0122   0.0122     0.0127

sigma^2 estimated as 1.004:  log likelihood = -7104.21,  aic = 14216.43
```

```
res <- arma_final$residuals
Box.test(res)
```

```

    Box-Pierce test

data:  res
X-squared = 0.15166, df = 1, p-value = 0.697
```

```
par(mfrow=c(1,2))
qqnorm(res)
abline(a=0, b=1, col="red")
hist(res, probability = TRUE)
lines(density(res))
x = seq(min(res),max(res), by=0.1)
lines(x, dnorm(x), col="red")
```



Finalement, on notre modèle est un ARIMA(2,1,0) (ou ARI(2,1)).

3.- Utiliser la fonction `auto.arima` de la librairie `forecast` pour comparer.

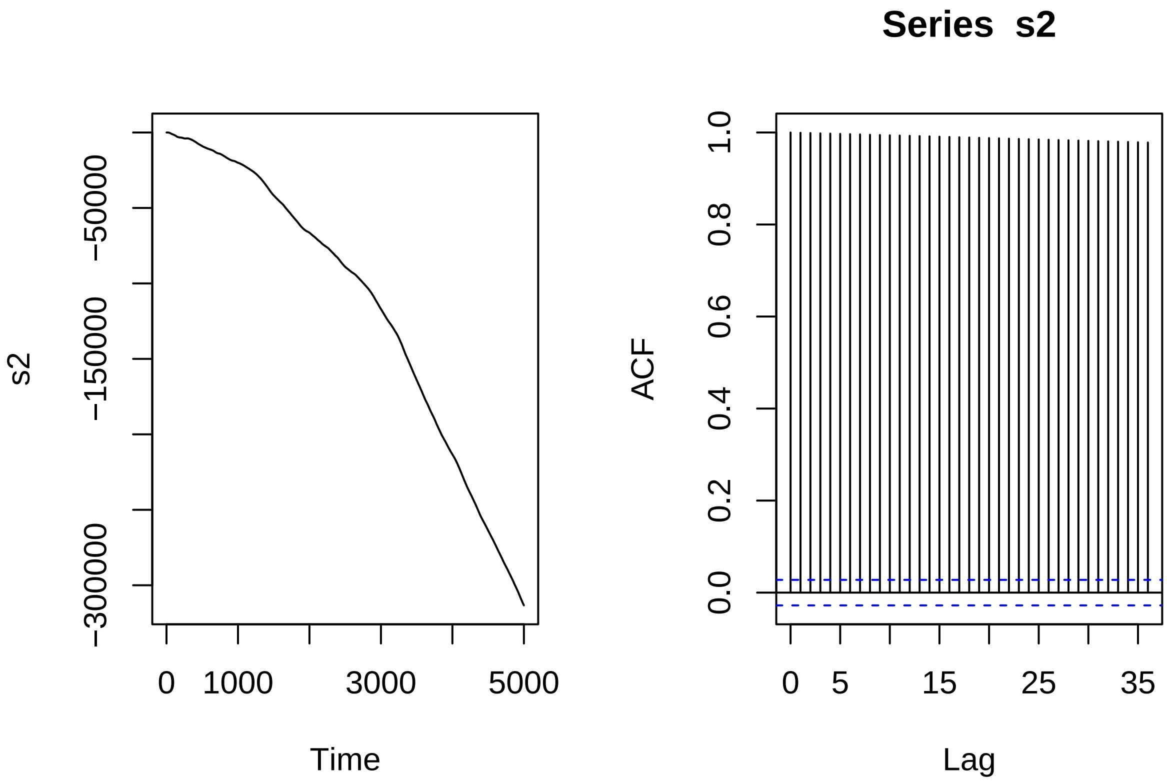
```
library(forecast)
arima.automatic <- auto.arima(s1)
arima.automatic
```

```
Series: s1
ARIMA(2,1,0) with drift

Coefficients:
         ar1      ar2    drift
      0.3936  -0.5068  -0.0286
s.e.  0.0122   0.0122   0.0127

sigma^2 = 1.005:  log likelihood = -7104.21
AIC=14216.43   AICc=14216.43   BIC=14242.49
```

L'algorithme a trouvé exactement le même resultat que nous.

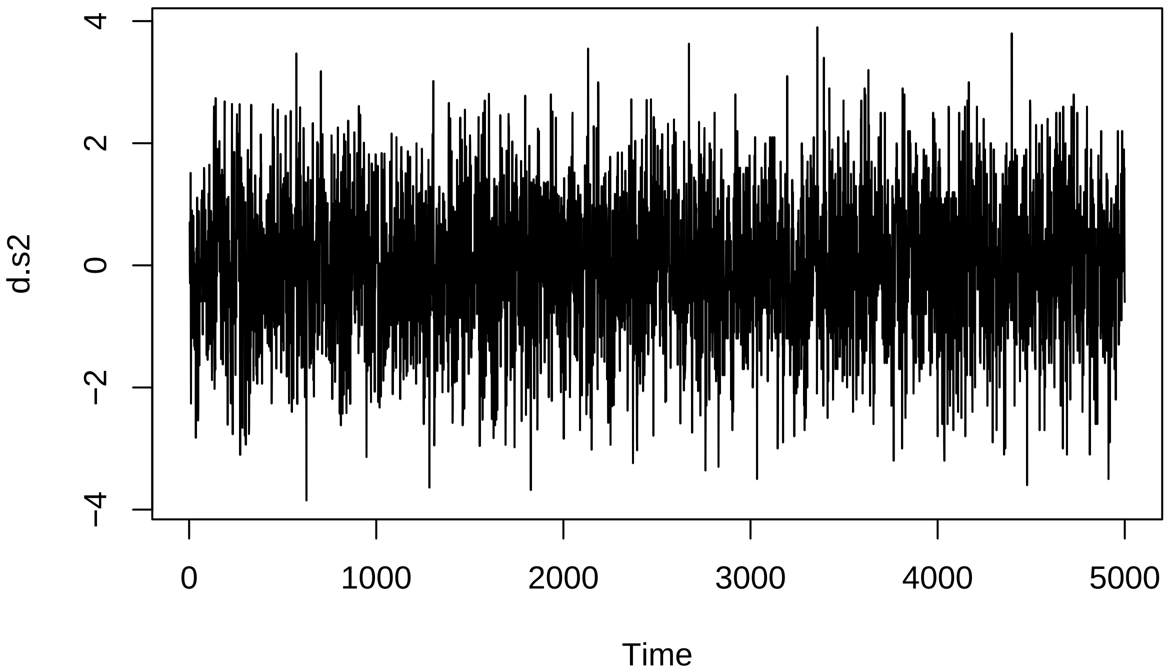
On procède de la même manière pour le deuxième tableau de données :

```r
s2 <- scan("serie2.dat")
s2 <- ts(s2, frequency = 1)
par(mfrow=c(1,2))
plot.ts(s2)
acf(s2)
```

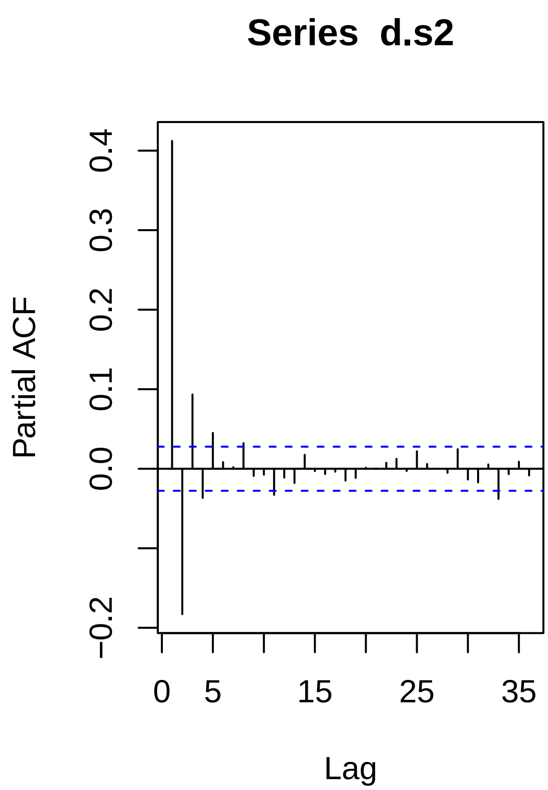


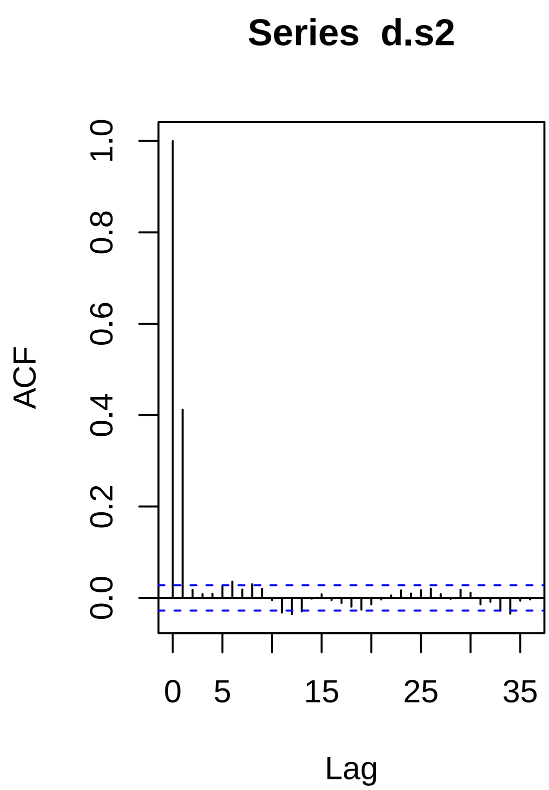
La non-stationarité est claire. Donc, on la différencie

```r
d.s2 <- diff(s2, differences = 2)# C'est équivalent à diff(diff(X_t)),
#i.e., à l'opérateur (I-B)^2(X_t).
plot.ts(d.s2)
```




```
par(mfrow=c(1,2))
pacf(d.s2)
acf(d.s2)
```





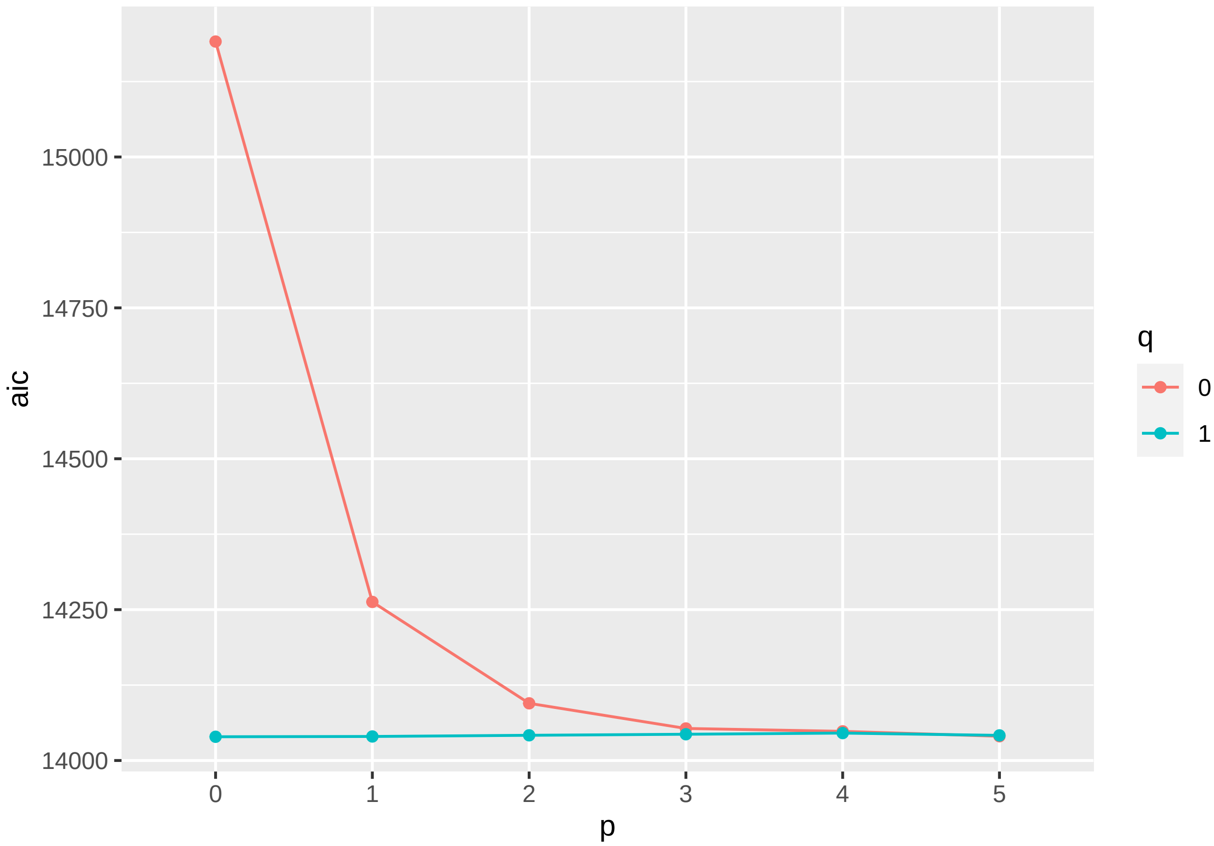
Ici, $p_{max} = 5$ et $q_{max} = 1$

```
p.max=5
q.max=1
Results.aic2 <- matrix(NA, ncol = p.max+1, nrow = q.max+1)
```

```r
Results.loglik2 <- matrix(NA, ncol = p.max+1, nrow = q.max+1)
Results.pvalue.res2 <- matrix(NA, ncol = p.max+1, nrow = q.max+1)
for(p in 0:p.max){
  for (q in 0:q.max){
    arma <- arima(d.s2, order = c(p,0,q))
    Results.aic2[(q+1),(p+1)] <- arma$aic
    Results.loglik2[(q+1),(p+1)] <- - arma$loglik
    Results.pvalue.res2[(q+1),(p+1)] <- Box.test(arma$residuals)$p.value
  }
}
```

```r
Results.aic.df <- data.frame(Results.aic2)
colnames(Results.aic.df) <- as.character(0:p.max)
Results.aic.g <- gather(Results.aic.df, key = "p", value = "aic")

Results.loglik.df <- data.frame(Results.loglik2)
colnames(Results.loglik.df) <- as.character(0:p.max)
Results.loglik.g <- gather(Results.loglik.df, key = "p", value = "loglik")

Results.pvalue.res.df <- data.frame(Results.pvalue.res2)
colnames(Results.pvalue.res.df) <- as.character(0:p.max)
Results.pvalue.res.g <- gather(Results.pvalue.res.df, key = "p", value = "p.value")

Results <- cbind(0:q.max, Results.aic.g, Results.loglik.g$loglik, Results.pvalue.res.g$p.value)
colnames(Results) <- c("q", "p", "aic", "loglik", "p.value")
Results$q <- as.factor(Results$q)
```

```r
ggplot(Results, aes(x=p, y= aic, group=q)) +
  geom_line(aes(color=q))+
  geom_point(aes(color=q)) #+ ylim(14000, 14100)
```



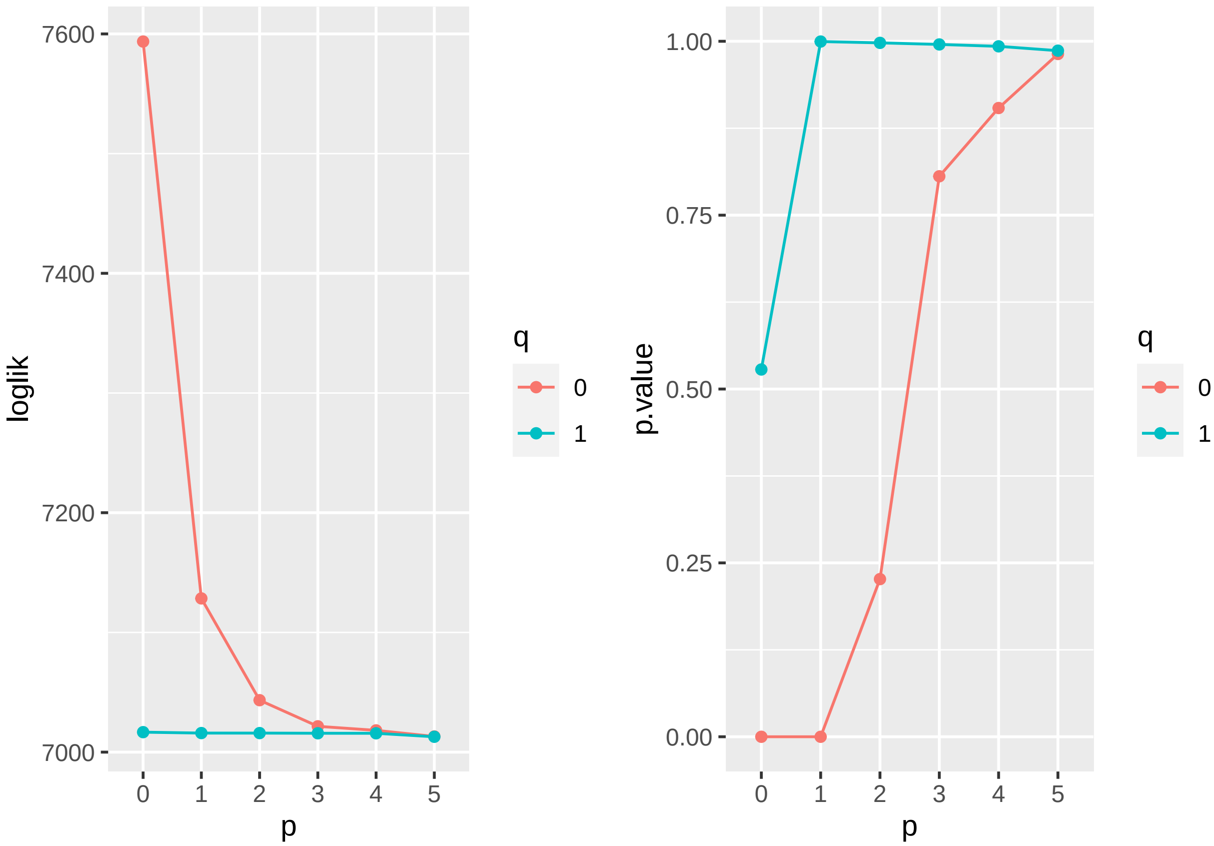
Le graphe montre que $q = 1$. Pour être sur du bon p, on calcule :

```
which(Results$aic[Results$q==1] == min(Results$aic[Results$q==1]))
```

```
[1] 1
```

Donc, $p = 0$. C'est-à-dire, le modèle retenu est un MA(1).

```
g1 <- ggplot(Results, aes(x=p, y= loglik, group=q)) +
  geom_line(aes(color=q))+
  geom_point(aes(color=q))# + ylim(7000, 7050)

g2 <- ggplot(Results, aes(x=p, y= p.value, group=q)) +
  geom_line(aes(color=q))+
  geom_point(aes(color=q))
plot_grid(g1,g2)
```



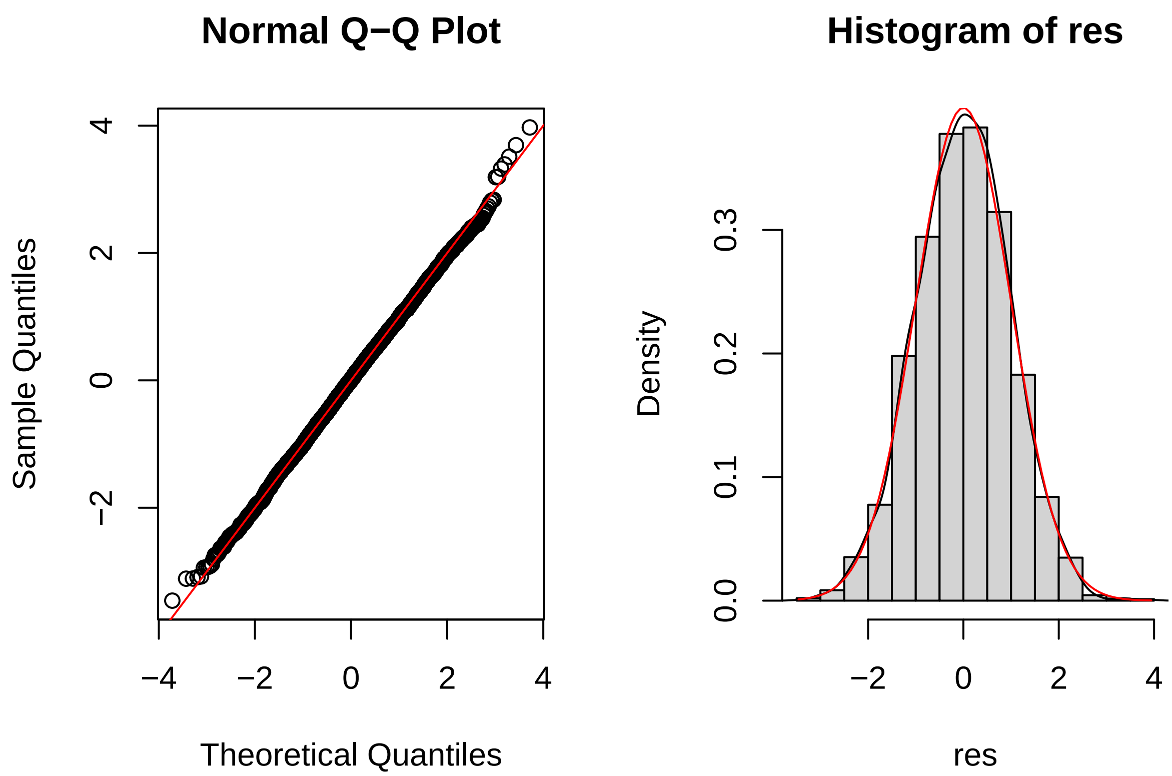
Conclusion : on a trouvé qu'un bon modèle pour cette série d'observations est un ARIMA(0,2,1).

On regarde en détail le modèle retenu :

```
arma_final <- arima(d.s2, order = c(0, 0, 1)) #MA(1)
res <- arma_final$residuals
Box.test(res)
```

```

    Box-Pierce test

data:  res
X-squared = 0.39809, df = 1, p-value = 0.5281
```

```
par(mfrow=c(1,2))
qqnorm(res)
abline(a=0,b=1, col="red")
hist(res, probability = TRUE)
lines(density(res))
x = seq(min(res),max(res), by=0.1)
lines(x, dnorm(x), col="red")
```



On regarde maintenant le résultat avec `auto.arima` :

```
arima.automatic2 <- auto.arima(s2)
arima.automatic2
```

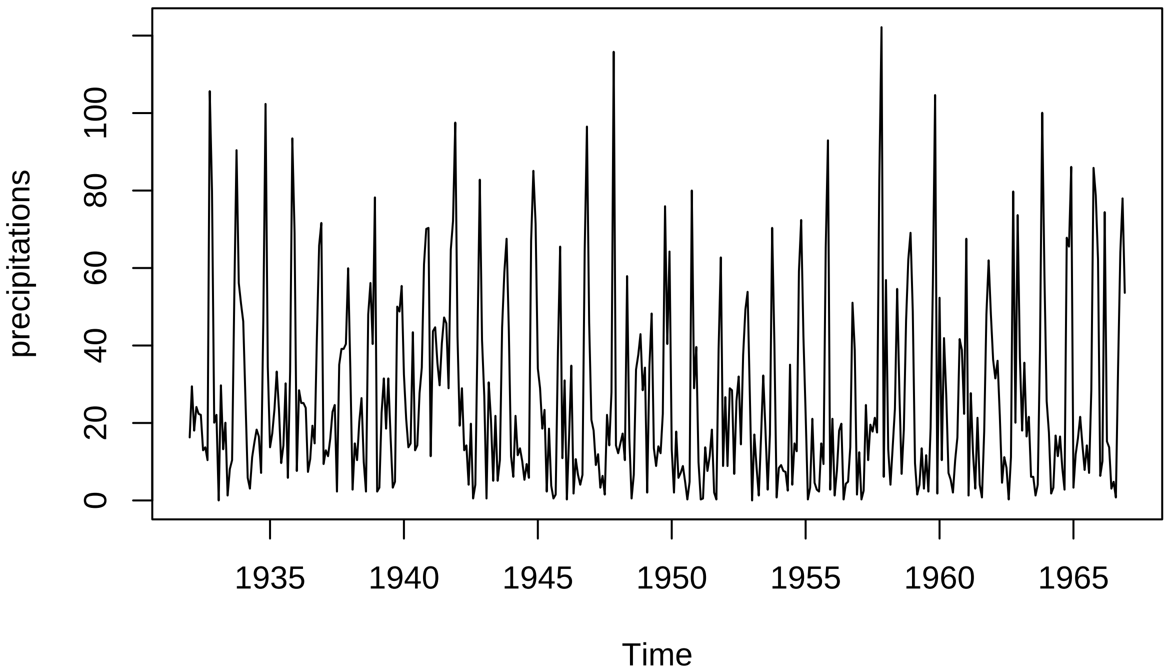
```
Series: s2
ARIMA(0,2,1)

Coefficients:
         ma1
      0.5009
s.e.  0.0120

sigma^2 = 0.9707:  log likelihood = -7017.1
AIC=14038.21   AICc=14038.21   BIC=14051.24
```

L'algorithme a trouvé exactement le même resultat que nous, un ARIMA(0,2,1).

Exercice 2

On s'intéresse aux précipitations mensuelles à San Fransisco entre 1932 et 1966, contenues dans le fichier `san_fran.txt`.

```
donnees <- scan("san_fran.txt")
precipitations <- ts(donnees, start=c(1932,1),end=c(1966,12),frequency=12)
plot.ts(precipitations)
```



1.- Utiliser la méthode de Box-Jenkins pour proposer une modélisation de type SARIMA pour cette série.

Rappel de la méthode de Box-Jenkins pour effectuer une modélisation SARIMA :

- Identifier la saisonnalité s avec l'autocorrélogramme. Vous pouvez également utiliser le spectogramme (non vu dans ce cours).

- Stationnariser la série, en commençant par D puis d.

- Déterminer des ordres p et q plausibles à l'aide de l'autocorrélogramme partiel et l'autocorrélogramme.

- Les ordres P et Q en regardant les ordre mulitples de s de ces autocorrélogrammes.

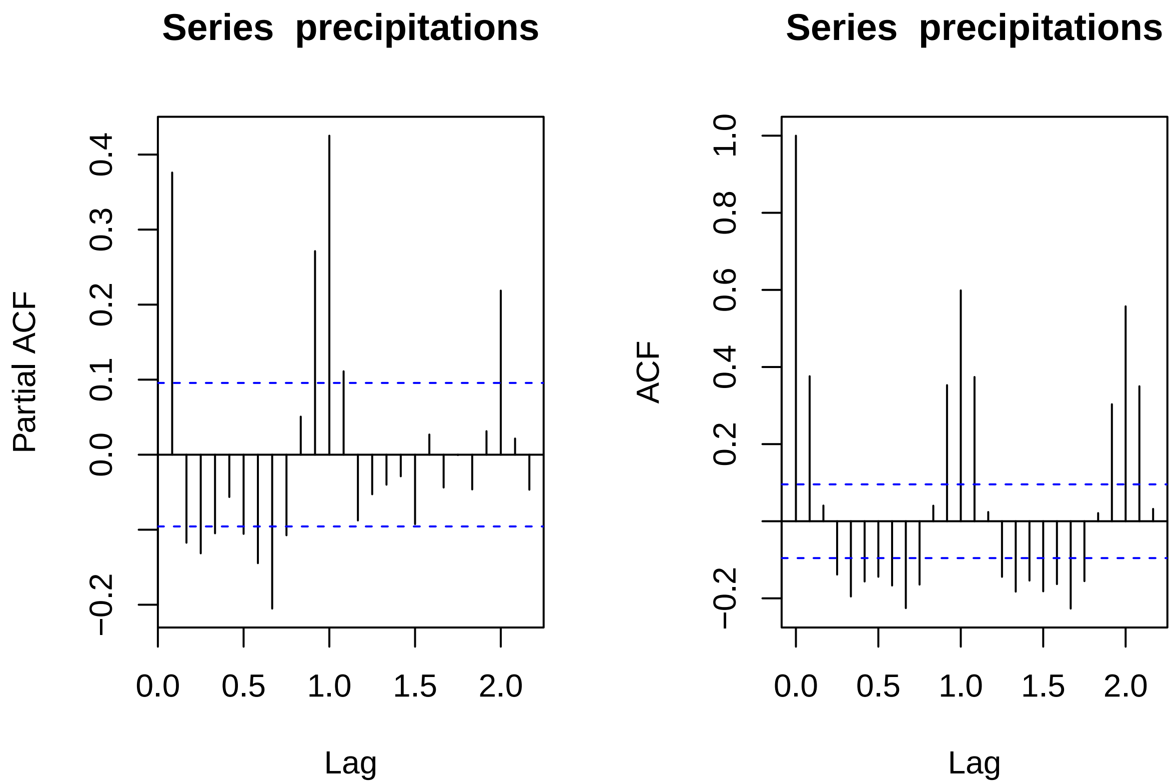
- Estimer les paramètres, si plusieurs modèles candidats, les départager par l'AIC ou le BIC.

- Valider ou non le modèle par un diagnostique des résidus (test, représentation graphique, autocorrélogramme).

- Confirmer votre choix en simulant de la prévision (échantillon test).

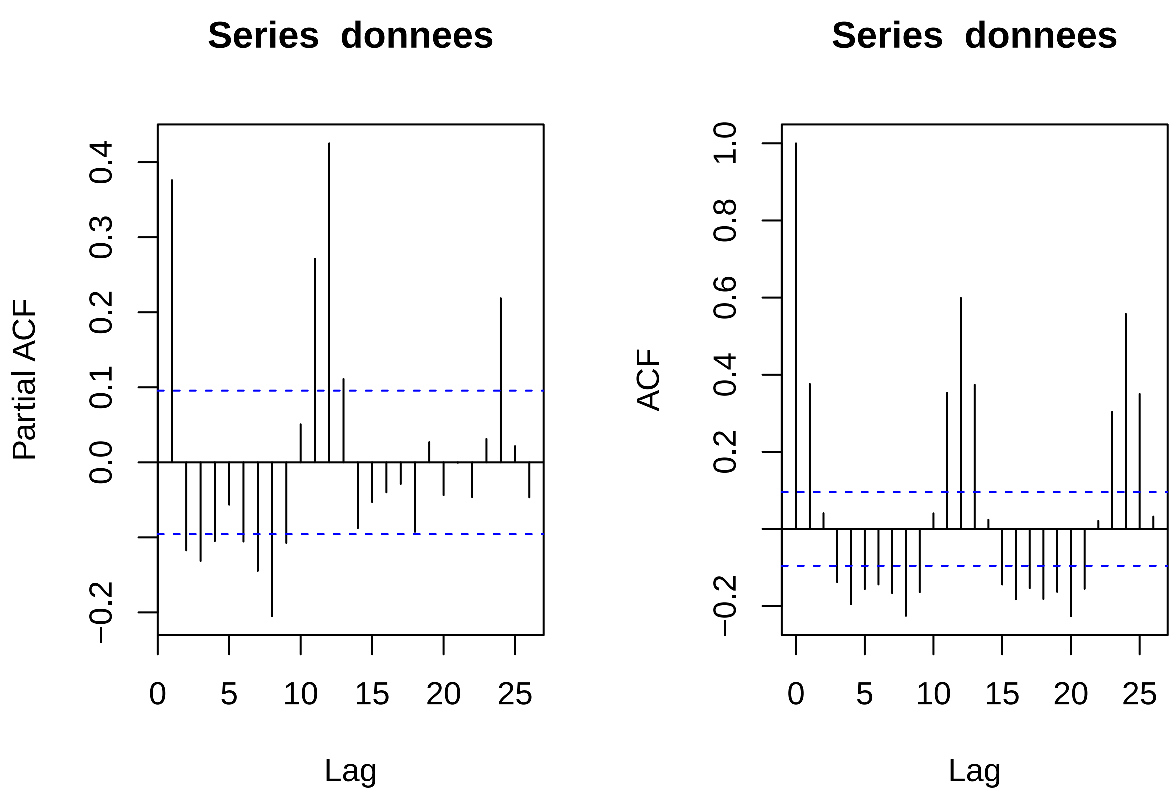
On commence alors par chercher la saisonnalité s (même si cela est évident en raison du type de données) :

```
par(mfrow=c(1,2))
pacf(precipitations)
acf(precipitations)
```



Ici, les pics sont à 1 (période). On pourrait les visualiser plus simplement de la manière suivante :

```
par(mfrow=c(1,2))
pacf(donnees)
acf(donnees)
```

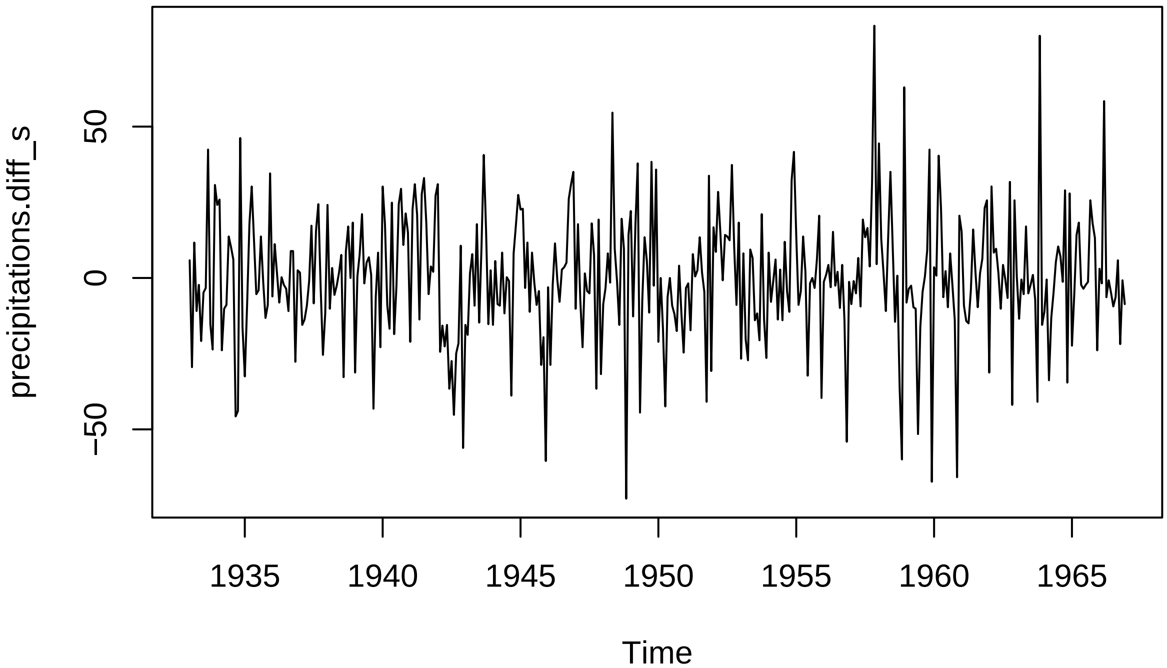


On

observe bien les pics aux multiples de 12.

On a alors une saisonnalité mensuelle. On applique le filtre $(I - B^{12})$ à la série pour la stationnariser :

```
precipitations.diff_s <- diff(precipitations, lag=12)
plot(precipitations.diff_s)
```



On teste la stationnarité de la série differentiée :

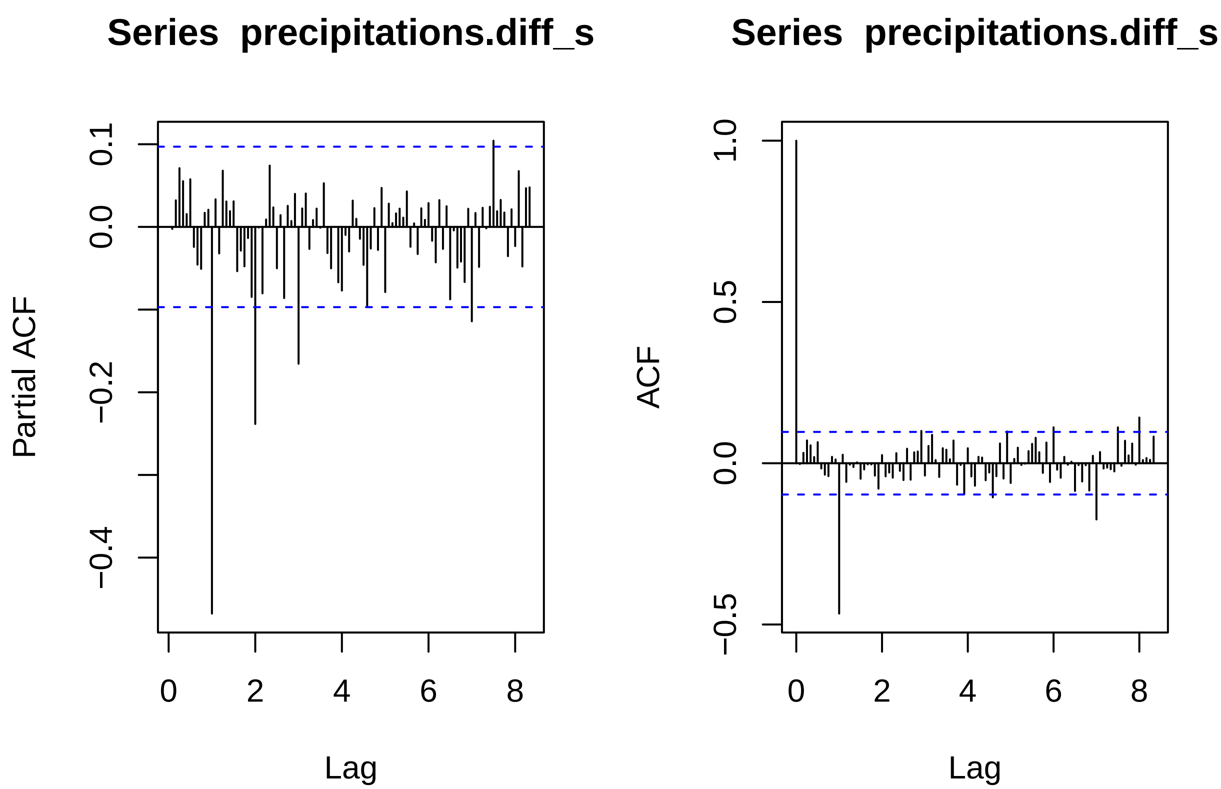
```
library(tseries)
adf.test(precipitations.diff_s)
```

```
    Augmented Dickey-Fuller Test

data:  precipitations.diff_s
Dickey-Fuller = -6.6089, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

On rejete alors l'hypothèse nulle (la série n'est pas stationnaire) à un risque inférieur à 1%. Notre série est alors stationnaire.

On regarde maintenant les acf et pacf de la série differentiée :

```
par(mfrow=c(1,2))
pacf(precipitations.diff_s, lag.max = 100)
acf(precipitations.diff_s, lag.max = 100)
```



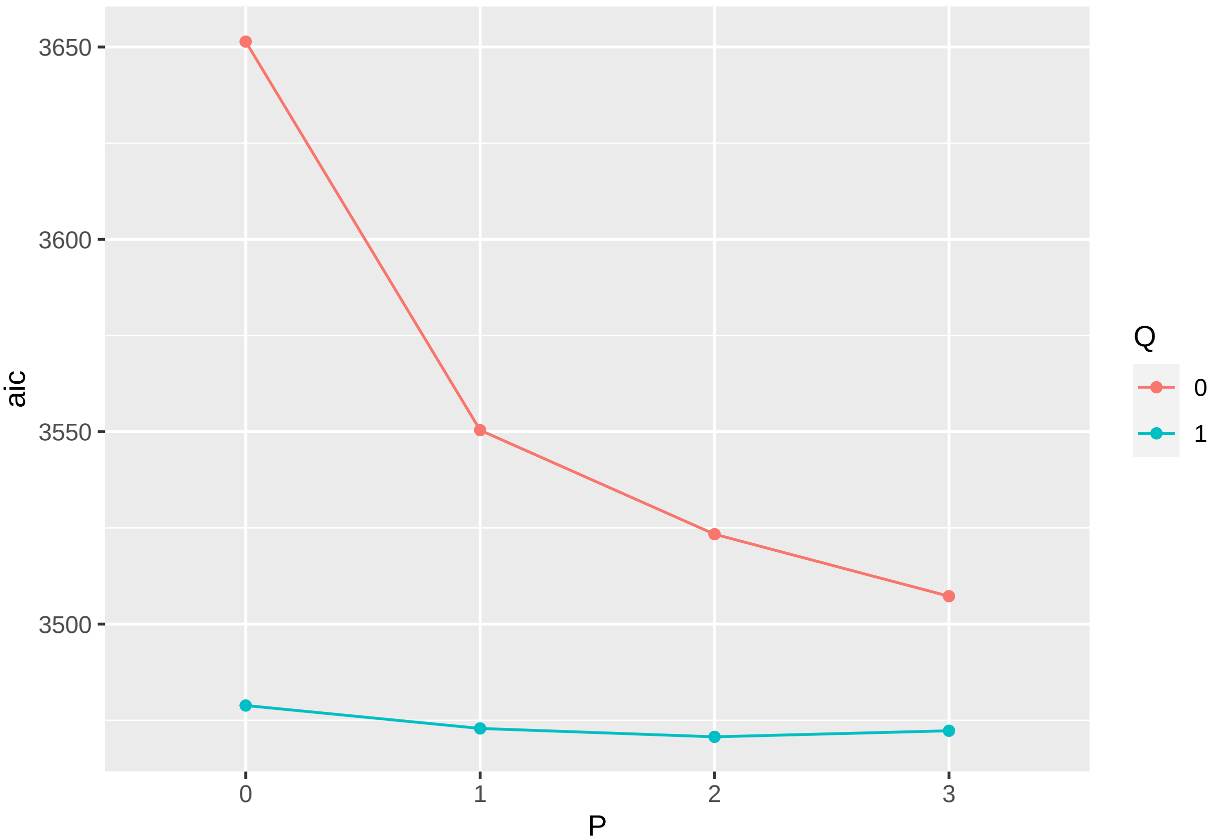
On n'obtient que des pics isolés aux multiples de 12. On utilise cette information pour estimer les parametrès P et Q. Soit $P_{max} = 3$ et $Q_{max} = 1$

```
P.max=3
Q.max=1
Results.aic.s <- matrix(NA, ncol = P.max+1, nrow = Q.max+1)
Results.loglik.s <- matrix(NA, ncol = P.max+1, nrow = Q.max+1)
Results.pvalue.res.s <- matrix(NA, ncol = P.max+1, nrow = Q.max+1)
for(P in 0:P.max){
  for (Q in 0:Q.max){
    sarima <- arima(precipitations.diff_s, order=c(0,0,0), seasonal=c(P,0,Q))
    Results.aic.s[(Q+1),(P+1)] <- sarima$aic
    Results.loglik.s[(Q+1),(P+1)] <- - sarima$loglik
    Results.pvalue.res.s[(Q+1),(P+1)] <- Box.test(sarima$residuals)$p.value
  }
}
```

```
Results.aic.df <- data.frame(Results.aic.s)
colnames(Results.aic.df) <- as.character(0:P.max)
Results.aic.g <- gather(Results.aic.df, key = "P", value = "aic")
#
Results.loglik.df <- data.frame(Results.loglik.s)
colnames(Results.loglik.df) <- as.character(0:P.max)
Results.loglik.g <- gather(Results.loglik.df, key = "P", value = "loglik")
#
Results.pvalue.res.df <- data.frame(Results.pvalue.res.s)
colnames(Results.pvalue.res.df) <- as.character(0:P.max)
Results.pvalue.res.g <- gather(Results.pvalue.res.df, key = "P",
                               value = "p.value")
#
```

```r
Results <- cbind(0:Q.max, Results.aic.g, Results.loglik.g$loglik,
                 Results.pvalue.res.g$p.value)
colnames(Results) <- c("Q", "P", "aic", "loglik", "p.value")
Results$Q <- as.factor(Results$Q)
```

```r
ggplot(Results, aes(x=P, y=aic, group=Q)) +
  geom_line(aes(color=Q))+
  geom_point(aes(color=Q)) #+ ylim(3450, 3500)
```



La graphique montre que nous devons choisir $P = 2$ et $Q = 1$.

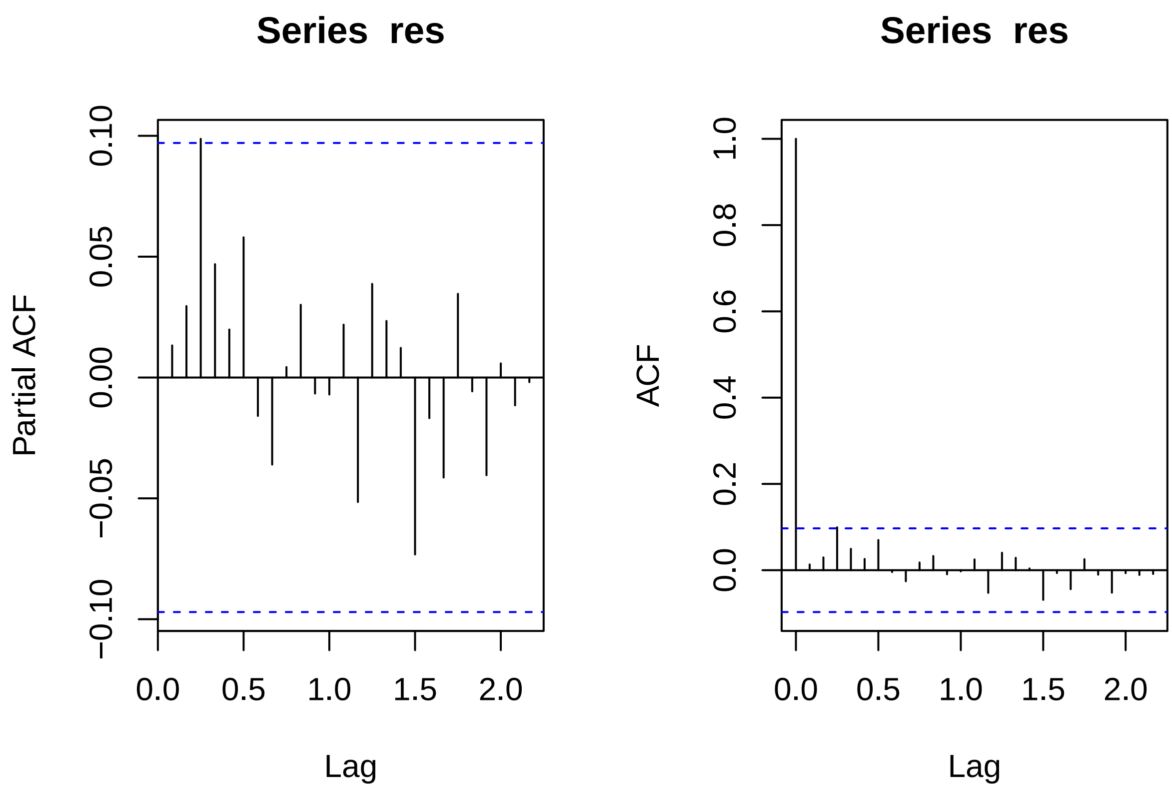
```r
sarima_final <- arima(precipitations.diff_s, order=c(0,0,0), seasonal=c(2,0,1))
sarima_final
```

```
Call:
arima(x = precipitations.diff_s, order = c(0, 0, 0), seasonal = c(2, 0, 1))

Coefficients:
        sar1    sar2     sma1  intercept
      0.1474  0.1063  -1.0000    -0.1253
s.e.  0.0514  0.0521   0.0367     0.1000

sigma^2 estimated as 258.4:  log likelihood = -1730.35,  aic = 3470.71
```

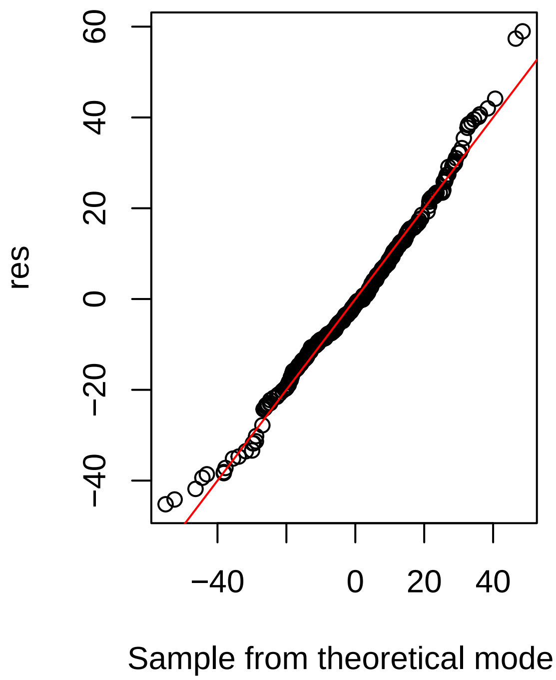
```r
res <- sarima_final$residuals
Box.test(res)
```

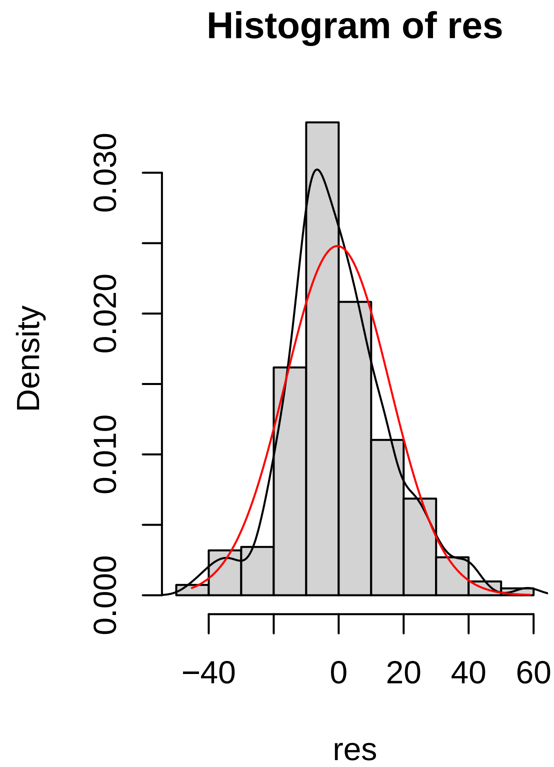
```
	Box-Pierce test

data:  res
X-squared = 0.071852, df = 1, p-value = 0.7887
```

```r
par(mfrow=c(1,2))
pacf(res)
acf(res)
```

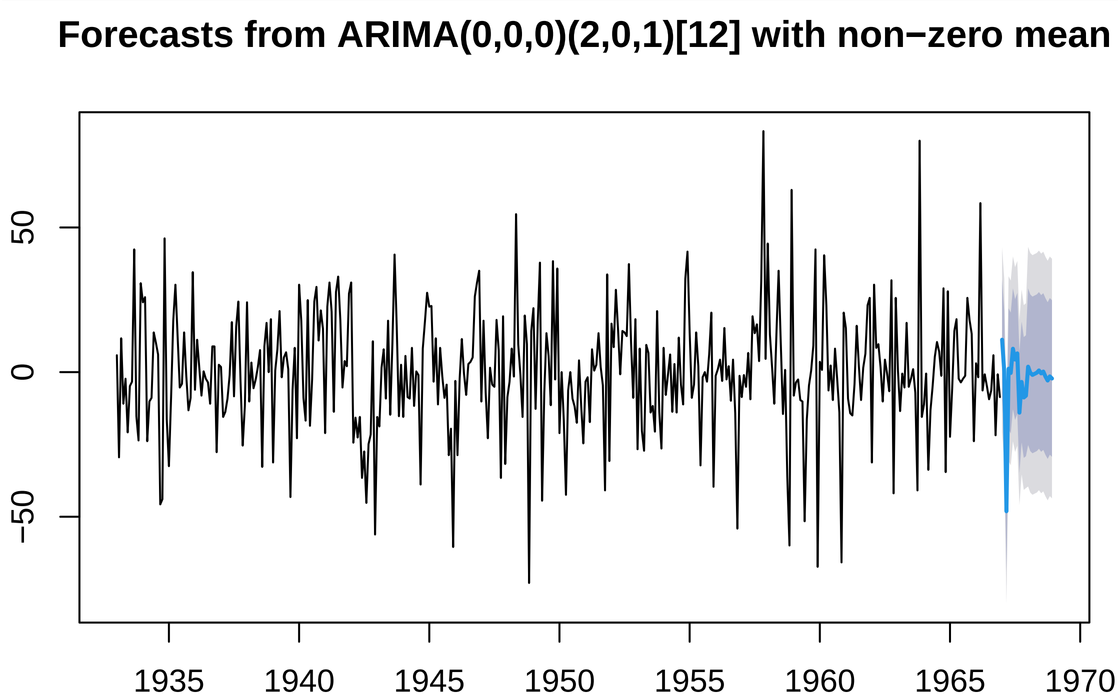



```r
set.seed(1234)
sample.norm <- rnorm(length(res), mean(res), sd(res))
par(mfrow=c(1,2))
qqplot(sample.norm, res, xlab = "Sample from theoretical model", ylab="res")
abline(a=0, b=1, col="red")
hist(res, probability = TRUE)
lines(density(res))
x = seq(min(res),max(res), by=0.1)
lines(x, dnorm(x, mean(res), sd(res)), col="red")
```





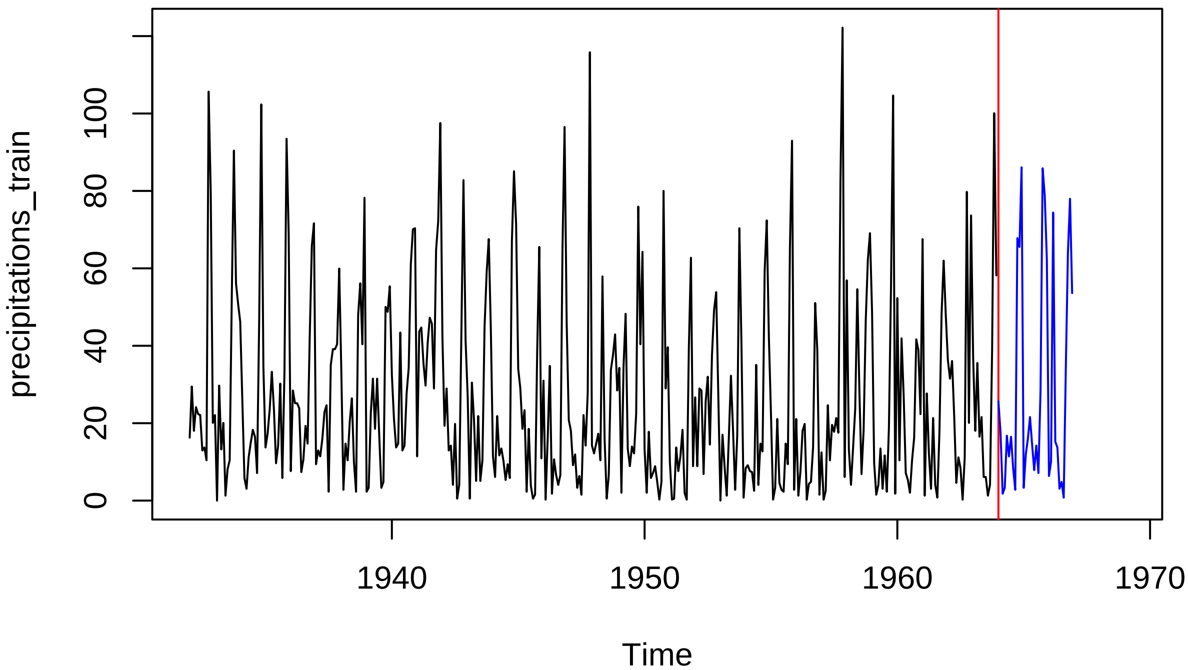

```r
library(forecast)
plot(forecast(sarima_final)) #Pour regarder la prévision à l'horizon
```



2.- Créer une série d'apprentissage contenant 32 années (91% des données), puis une série test avec le reste.

```r
# Apprentissage sur série réduite (fin en 63 au lieu de 66)
precipitations_train <- ts(donnees,start=c(1932,1),end=c(1963,12),frequency=12)
# série de test: le reste de la série
```

```
precipitations_test <- ts(donnees[385:420],start=c(1964,1),end=c(1966,12),frequency=12)
# on trace les deux parties côte à côte
plot(precipitations_train, xlim=c(1932,1969))
abline(v=1964, col="red")
lines(precipitations_test,col="blue")
```



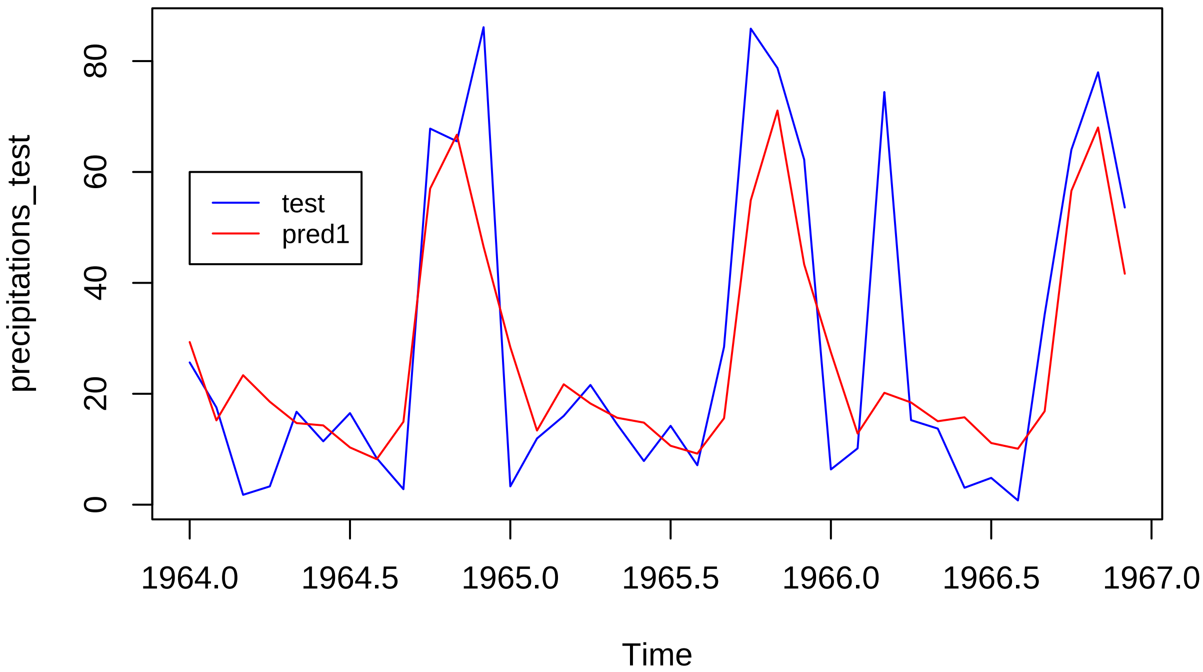
3.- Utiliser le modèle choisi pour prédire sur les données de test.

- Modèle choisi SARIMA$(0,0,0) \times (2,1,1)_{12}$:

```
mod1 <- arima(precipitations_train, order=c(0,0,0), seasonal=c(2,1,1))
```

On peut calculer la prédiction des 3 prochaines années (36 valeurs) comme suit :

```
pred1 <- predict(mod1,n.ahead=36)
# ou p1 <- forecast(modele_1, h=36)
plot(precipitations_test, col="blue")
lines(pred1$pred, col="red")
legend(1964, 60, legend=c("test", "pred1"),
       col=c("blue", "red"), lty=1, cex=0.8)
```

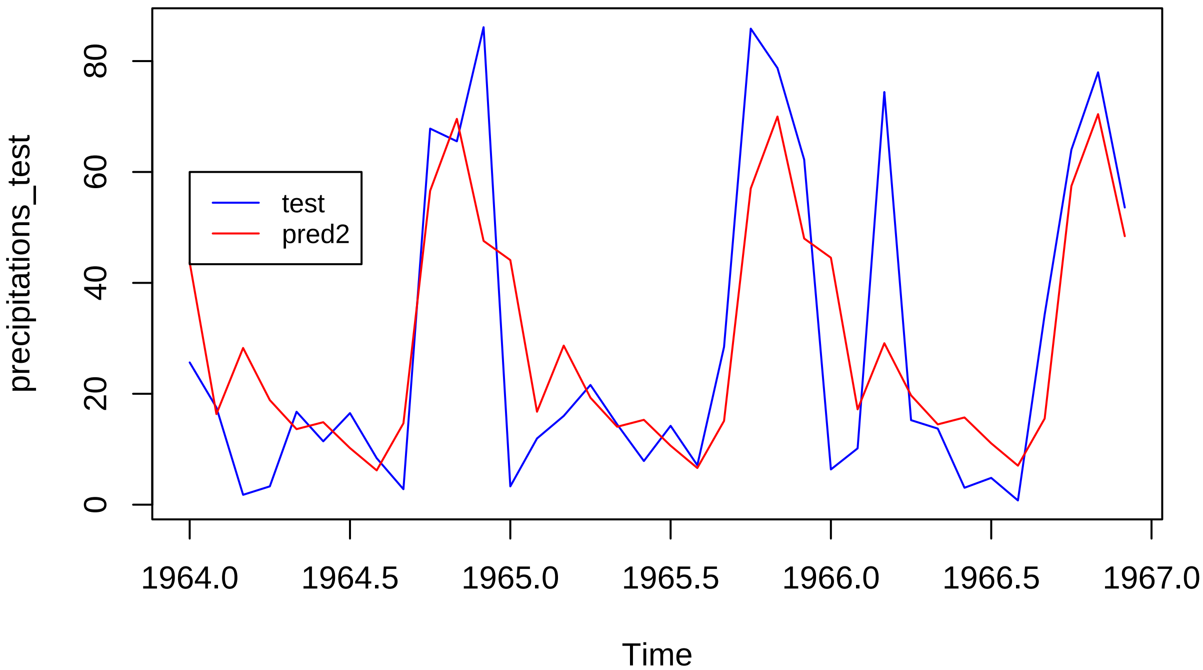


4.- Même question avec un lissage exponentiel de Holt-Winters (voir la fin du Chapitre 5 sur moodle) puis en faisant appel à la fonction `auto.arima` de la librairie `forecast`.

```
## Idem via Holt-Winters
mod2 <- HoltWinters(precipitations_train)
summary(mod2)
```

```
             Length Class  Mode     
fitted       1488   mts    numeric  
x             384   ts     numeric  
alpha           1   -none- numeric  
beta            1   -none- numeric  
gamma           1   -none- numeric  
coefficients   14   -none- numeric  
seasonal        1   -none- character
SSE             1   -none- numeric  
call            2   -none- call     
```

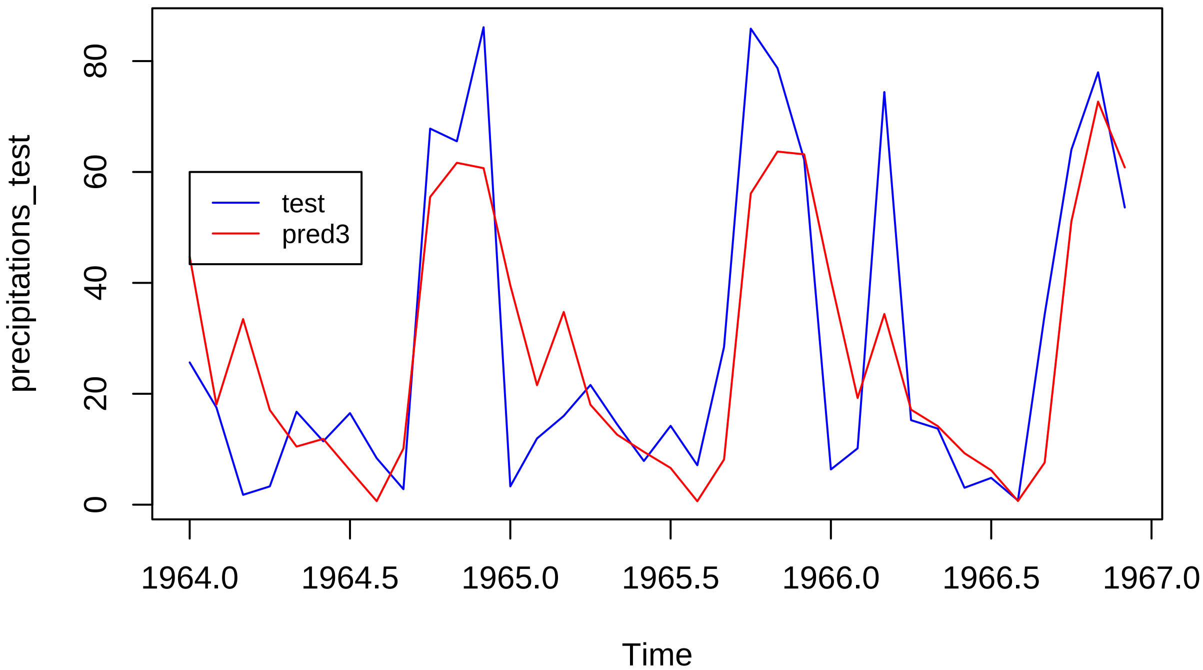
```
pred2 <- predict(mod2, n.ahead=36)
plot(precipitations_test, col="blue")
lines(pred2, col="red")
legend(1964, 60, legend=c("test", "pred2"),
       col=c("blue", "red"), lty=1, cex=0.8)
```




```
# et avec la fonction auto.arima
mod3 <- auto.arima(precipitations_train)
summary(mod3)
```

```
Series: precipitations_train
ARIMA(0,0,1)(2,1,0)[12] with drift

Coefficients:
          ma1      sar1      sar2    drift
      -0.0108   -0.6204   -0.2710  -0.0061
s.e.   0.0510    0.0508    0.0521   0.0415

sigma^2 = 327.7:  log likelihood = -1605.73
AIC=3221.46   AICc=3221.62   BIC=3241.05

Training set error measures:
                      ME     RMSE      MAE  MPE MAPE      MASE         ACF1
Training set -0.04091511 17.72204 13.27102 -Inf  Inf 0.8385827 0.0009082294
```

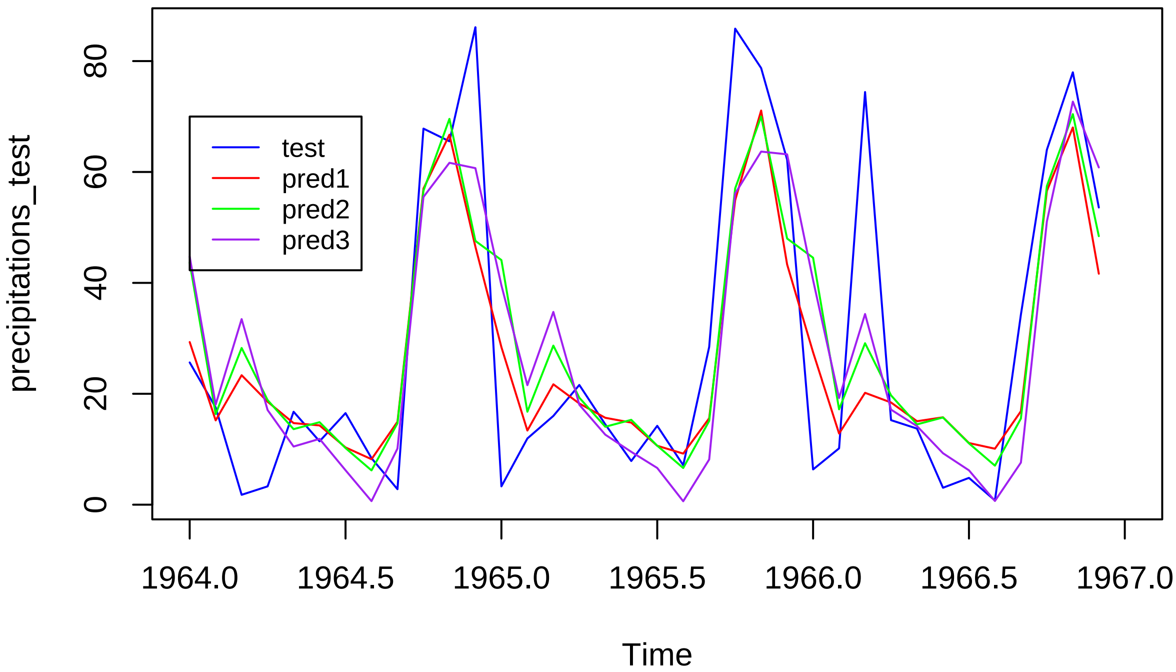
```
pred3 <- forecast(mod3, h=36)
plot(precipitations_test, col="blue")
lines(pred3$mean, col="red")
legend(1964, 60, legend=c("test", "pred3"),
       col=c("blue", "red"), lty=1, cex=0.8)
```



5.- Comparer les trois méthodes graphiquement puis à l'aide de l'erreur quadratique moyenne.

On regarde les trois prédictions sur la même fenetre pour comparer :

```
#plot(precipitations_train,xlim=c(1932,1969))
#abline(v=1964, col="red")
plot(precipitations_test, col="blue", xlim=c(1964,1967))
lines(pred1$pred, col="red")
lines(pred2, col= "green")
lines(pred3$mean, col="purple")
# lwd = line width
legend(1964, 70, legend=c("test", "pred1", "pred2", "pred3"),
       col=c("blue", "red", "green", "purple"), lty=1, cex=0.8)
```



On compare maintenant les erreurs quadratiques moyenne :

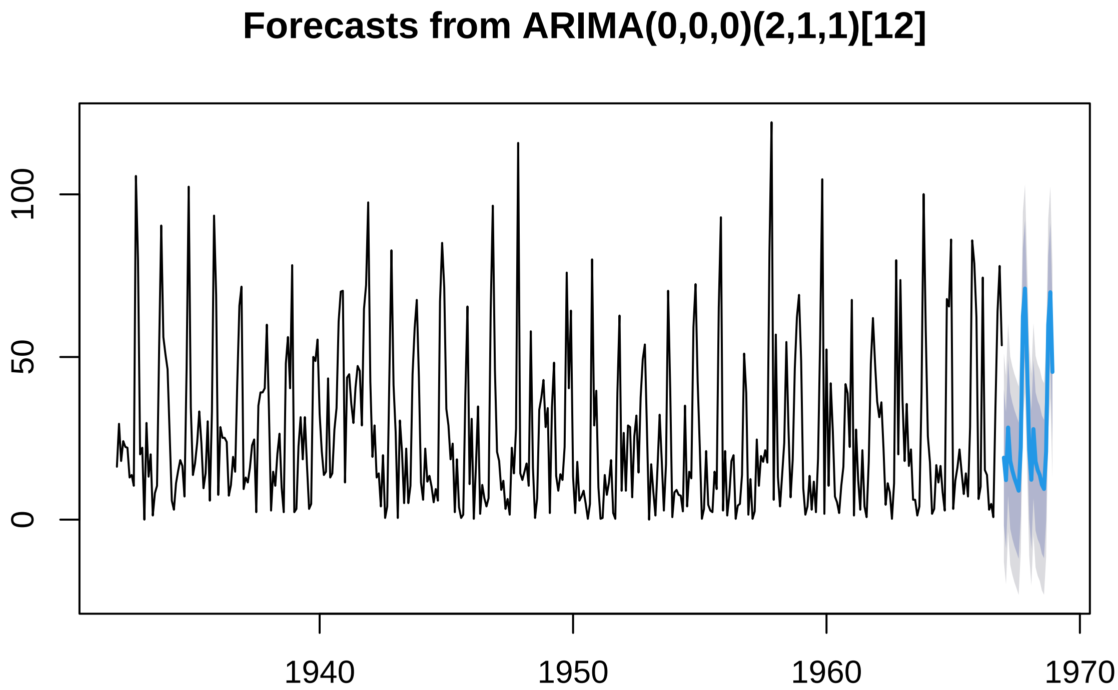
```
e1 = (sqrt(mean((pred1$pred - precipitations_test)^2)))
e2 = (sqrt(mean((pred2 - precipitations_test)^2)))
e3 = (sqrt(mean((pred3$mean - precipitations_test)^2)))
print(c(e1,e2,e3))
```

```
[1] 15.96924 17.17103 16.57343
```

Le premier modèle semble être le meilleur (par rapport à l'EQM).

6.- Faire de la vraie prédiction avec le meilleur modèle.

```
# prévisions futures d'après le modèle retenu
modele_retenu=arima(precipitations, order=c(0,0,0), seasonal=c(2,1,1))
plot(forecast(modele_retenu))
```


Forecasts from ARIMA(0,0,0)(2,1,1)[12]
100
50
0
1940
1950
1960
1970