Compte rendu du devoir Séries Temporelles

Nom et Prénom des étudiants du groupe :

```
- HO : An :
```

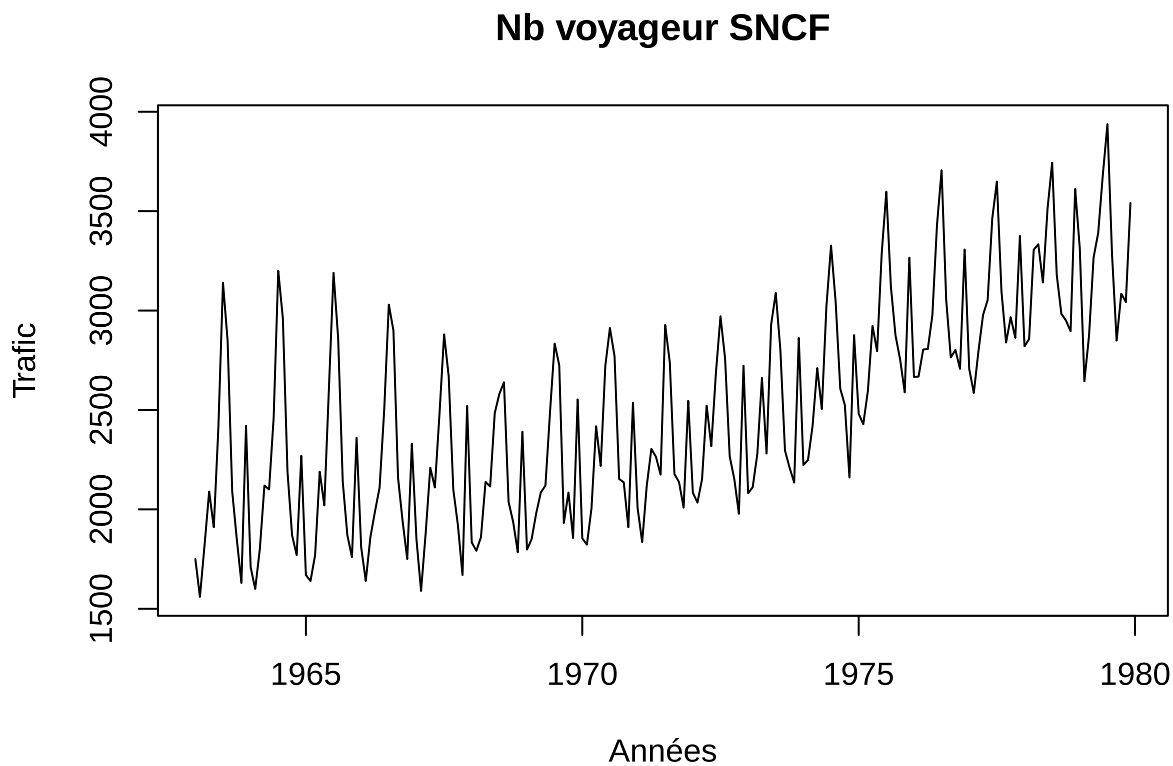
Instructions L'objectif de ce devoir serait d'effectuer l'analyser sur la série temporelle en étudiant 2 jeux de données : SNCF et numéro d'immatriculation en France

I. SCNF

Chargement et visualisation des données

```
sncf=read.table("http://freakonometrics.free.fr/sncf.csv",header=TRUE,sep=";")
train=as.vector(t(as.matrix(sncf[,2:13])))
X=ts(train,start = c(1963, 1), frequency = 12)

### on utlise les données de 1964 à 1979 pour l'entrainement et ceux de 1980 pour évaluer la qualité de
X.train = window(X,end=c(1979,12))
X.test <- window(X,start=1980)
plot(X.train,xlab='Années',ylab='Trafic',main="Nb voyageur SNCF")
```

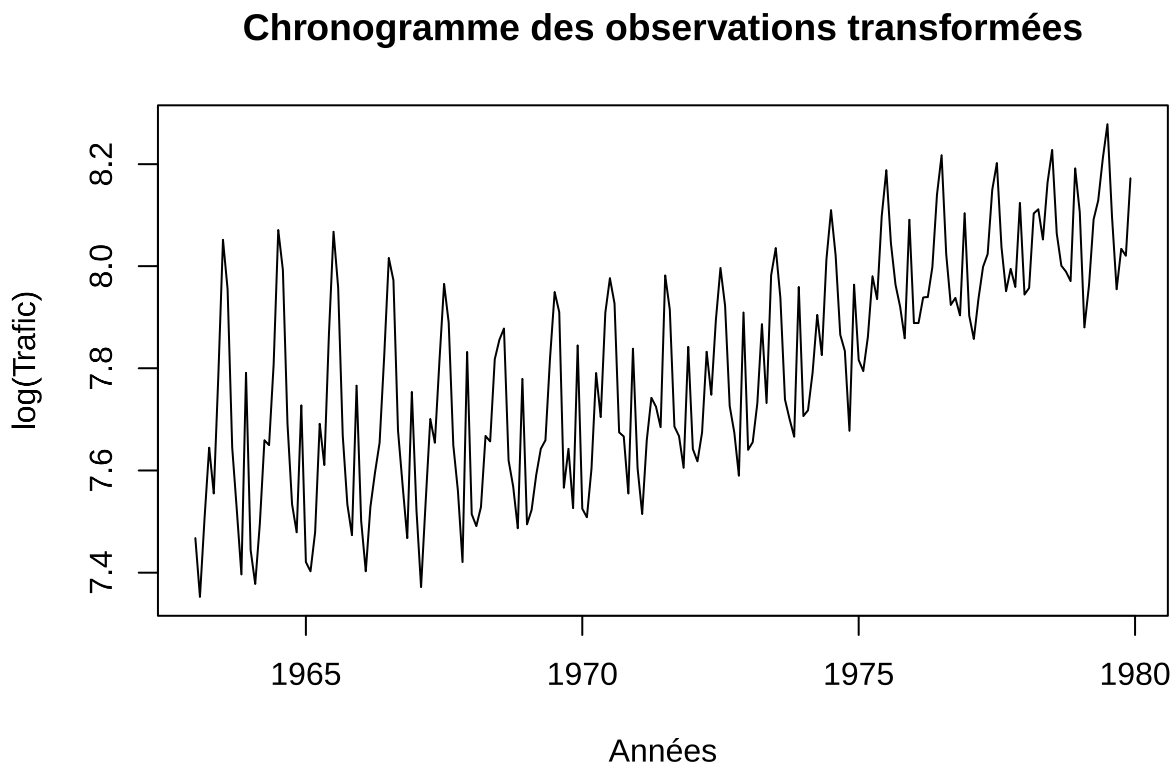


En réalité, on peut considérer que ce jeu de données devrait être une série chronologique, mensuelle, comportant une forte saisonalité

Saisonalité, tendance et résidus.

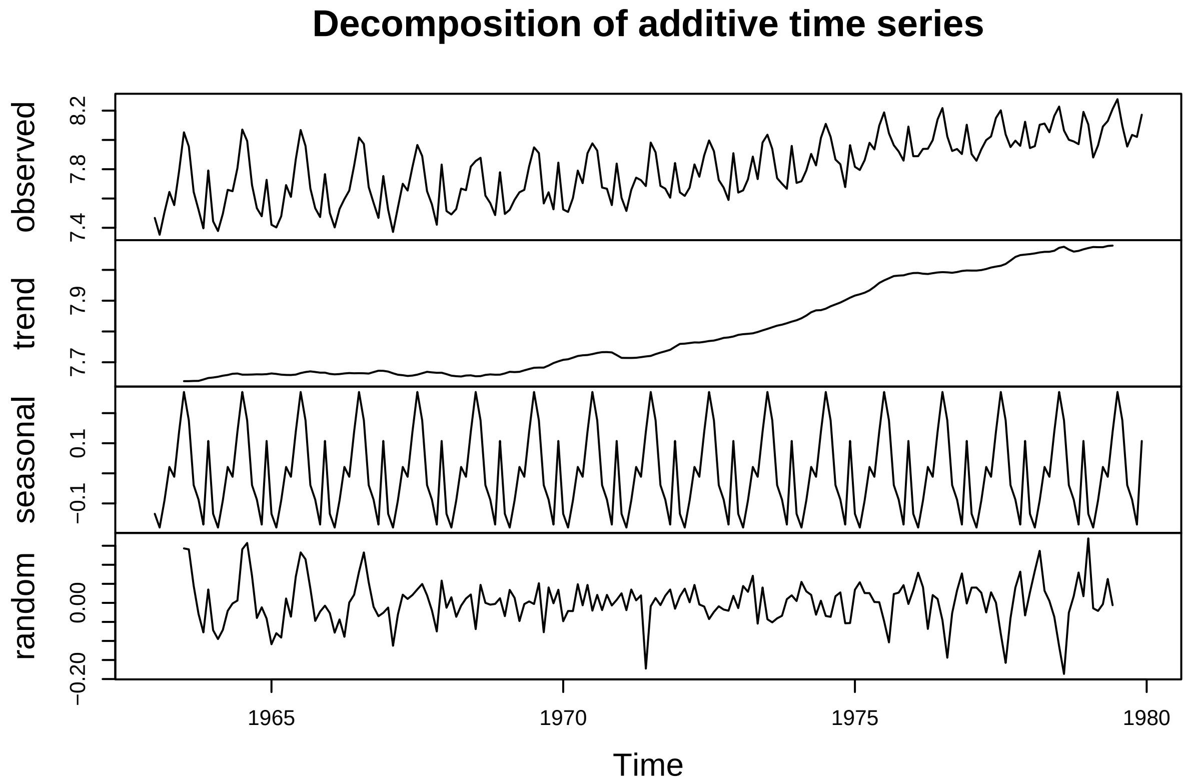
On va continuer à analyser avec le code R pour vérifier nos hypothèse. On trouve que la moyenne des oscillations varie de façon croissantes et non constante avec le temps, ce qui peut être une cause de non stationnarité de la série temporelle. Par ailleurs, l'amplitude des oscillations semble décroitre avec le temps, ce qui peut être un problème pour le choix et la forme du modèle (multiplicatif ou additif). Pour cela, nous analyserons la série du logarithme des observations de cet échantillon.

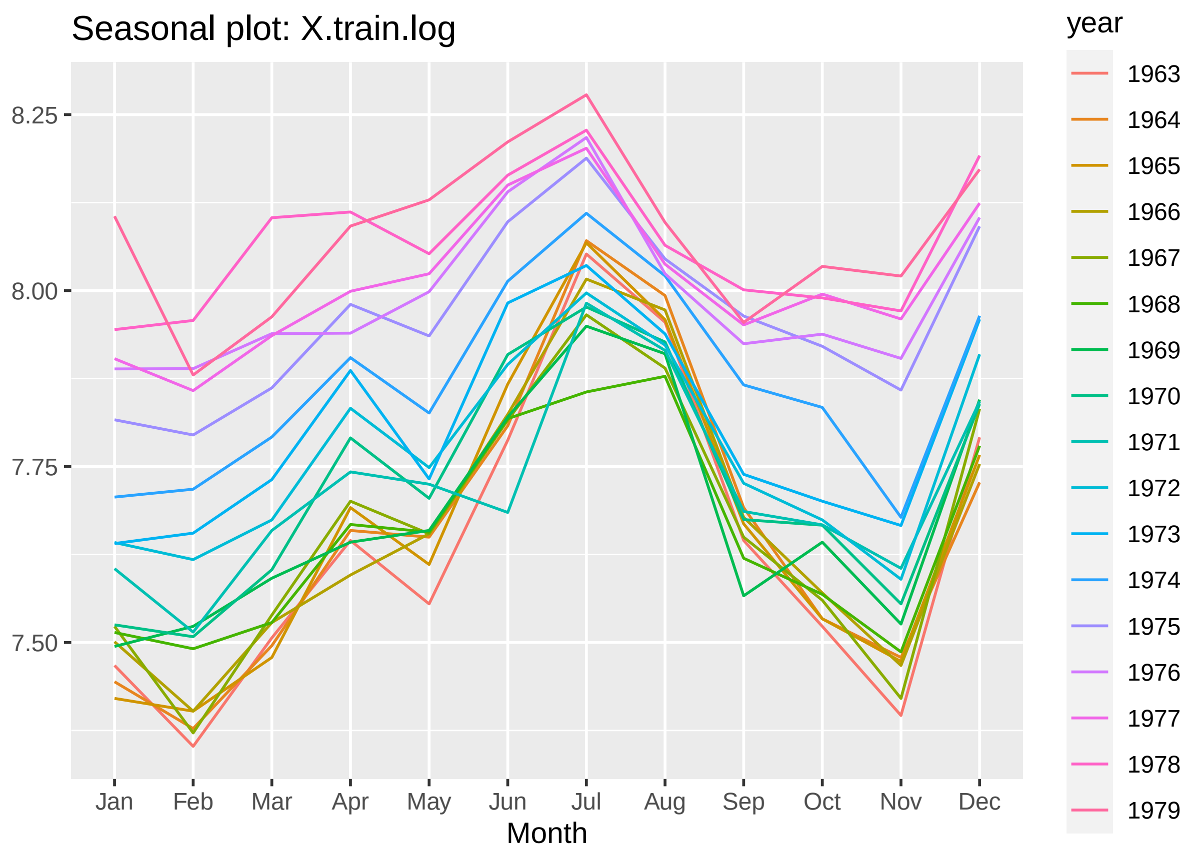
```
X.train.log = log(X.train)
X.test.log = log(X.test)
plot(X.train.log,xlab='Années',ylab='log(Trafic)',main="Chronogramme des observations transformées")
```



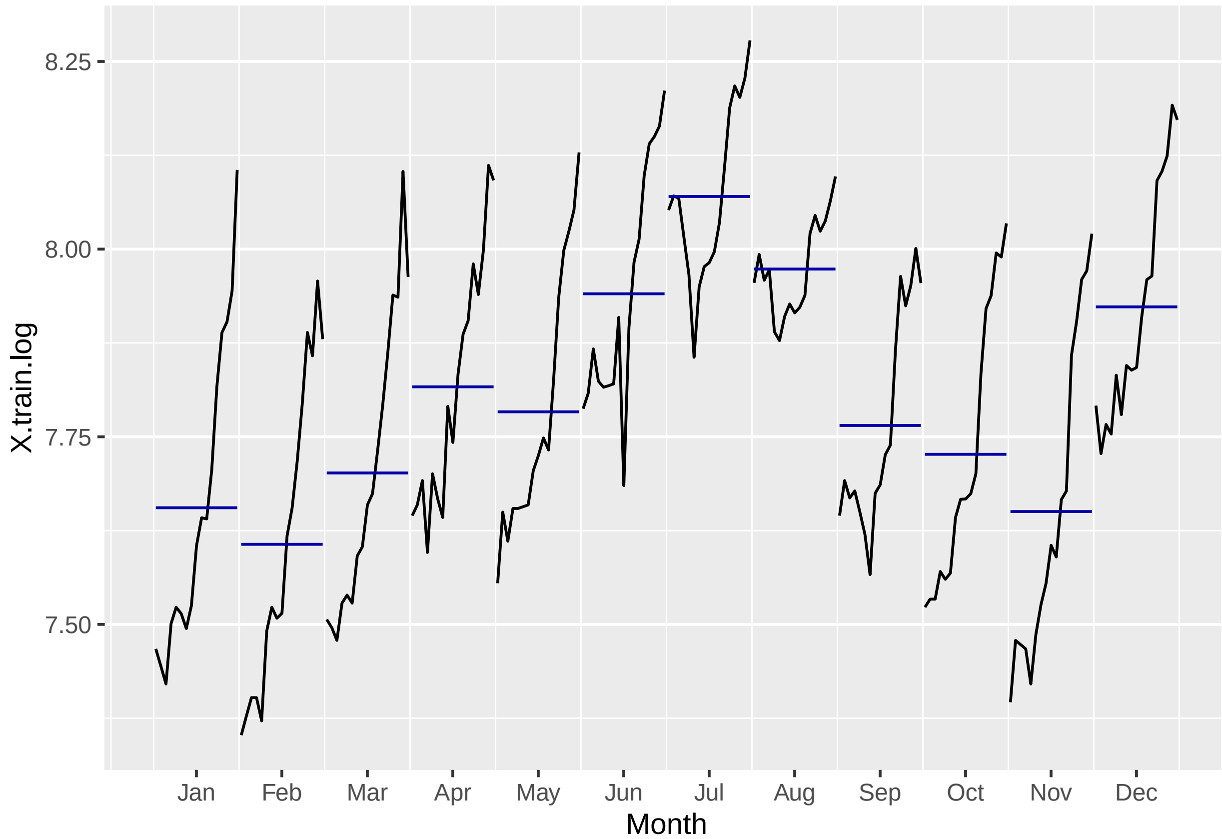
A partir de l'analyse de la figurenous permet de tirer les conclusions suivantes : Effectivement, l'amplitute semble constant à partir des annés 1968.Le modèle additif semble mieux adapté car les oscillations semblent varier entre 2 courbes parallèles

```
plot(decompose(X.train.log,type='additive'))
forecast::ggseasonplot(X.train.log)
```

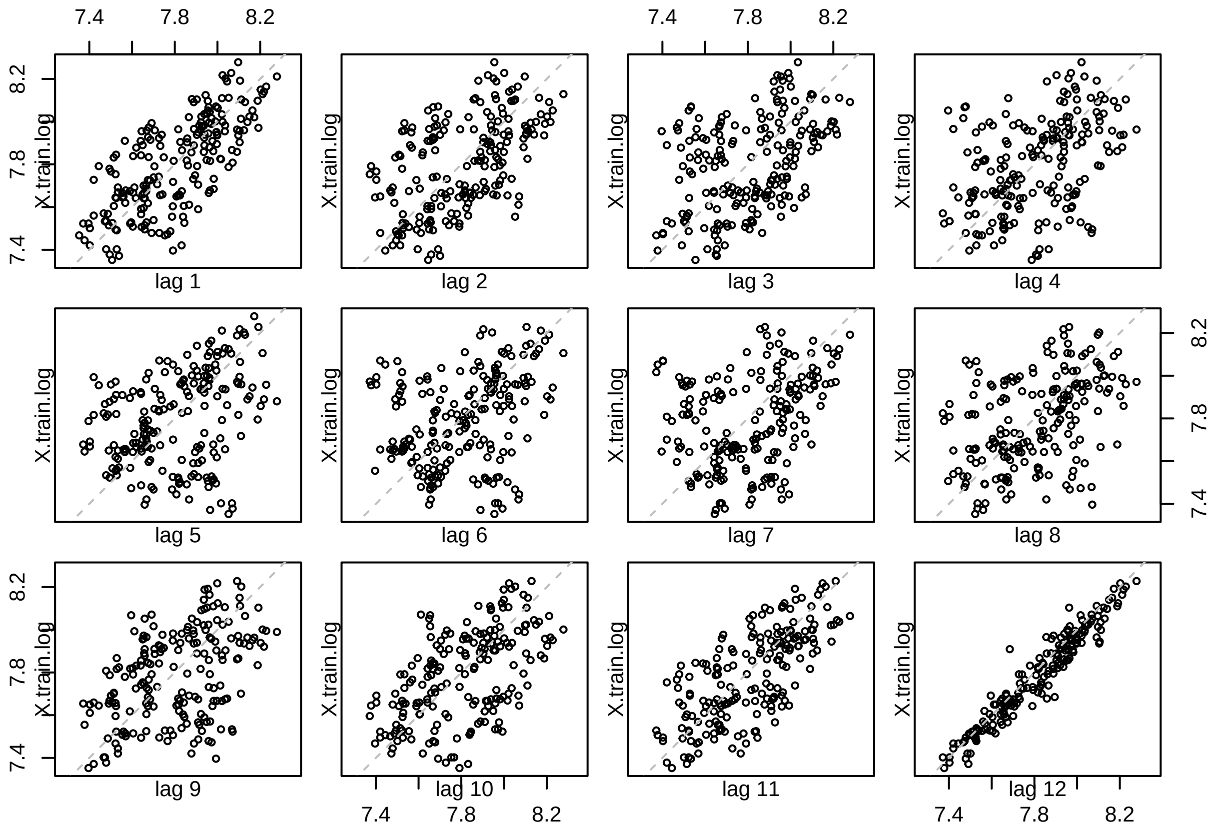

Decomposition of additive time series
observed
7.4
7.8
8.2
trend
7.7
7.9
seasonal
−0.1
0.1
random
−0.20
0.00
1965
1970
1975
1980
Time




```
forecast::ggsubseriesplot(X.train.log)
```




```
lag.plot(X.train.log,lags=12,layout=c(3,4),do.lines=FALSE)
```



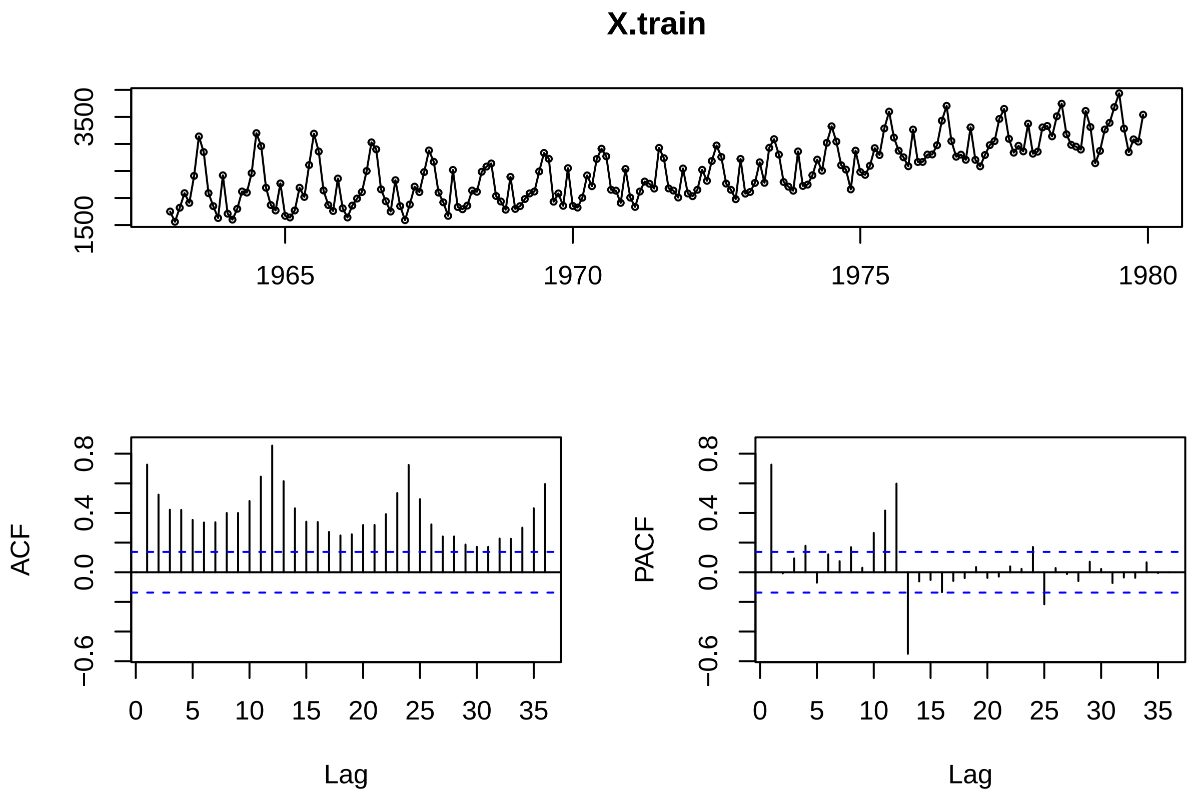
Sur les graphs, on observe bien une tendance générale croissante du nb de voyageur, la saisonalité marquée sur l'année et les résidus variées avec le temps. On constate également qu'il y a 2 pics en Décembre et en Juillet, ce dernier s'explique par les saisons de vacances d'été et d'hiver, notamment une forte corrélation au lag 12 et une petite corrélation au lag 1 - cela veut dire que la plupart du voyageurs part plutôt en vacances en Décembre. Par ailleurs, Le processus n'est clairement pas stationnaire.

On chercher donc à ramener notre série par différentiation à un processus stationnaire, que nous modéliserons.

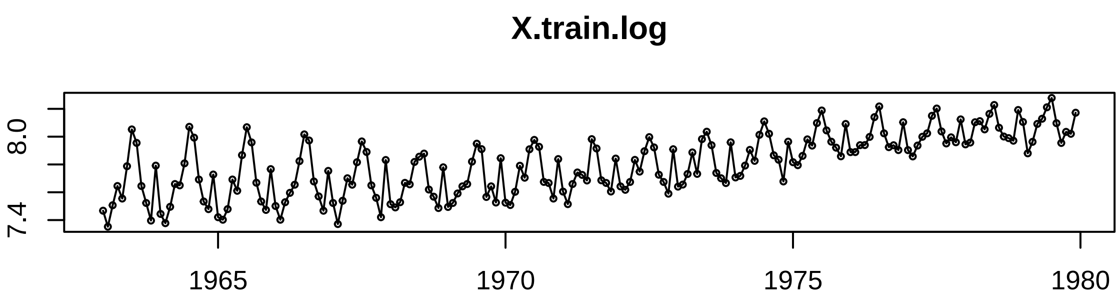
Différentiation

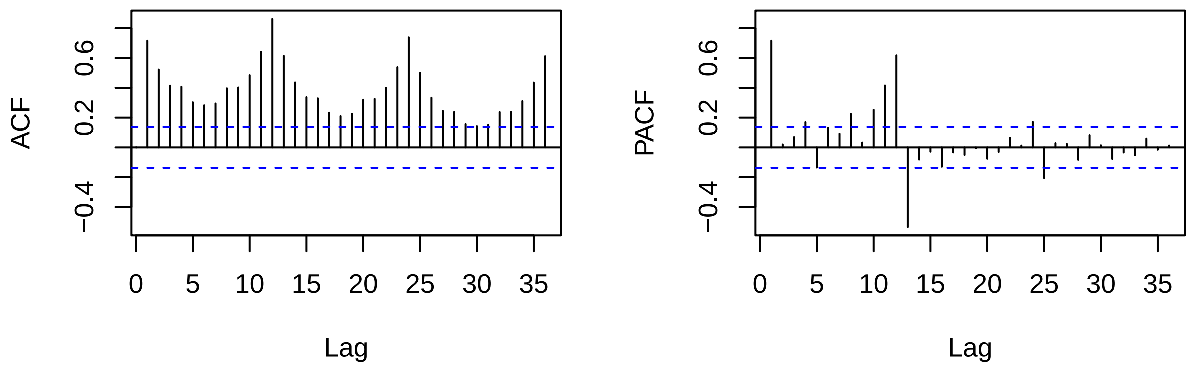
On affiche le graph de ACF et PACF du jeu de données

```
library('forecast')
tsdisplay(X.train)
```




```
tsdisplay(X.train.log)
```





même transformé en logarithme, le graph représente toujours un pic au lag 12.Nous observons une quasi non-significativité des auto-corrélations et auto-corrélations partielle. On va passer le test Ljung-Box pour savoir si la série temporelle entière peut être différenciée d'un bruit blanc.

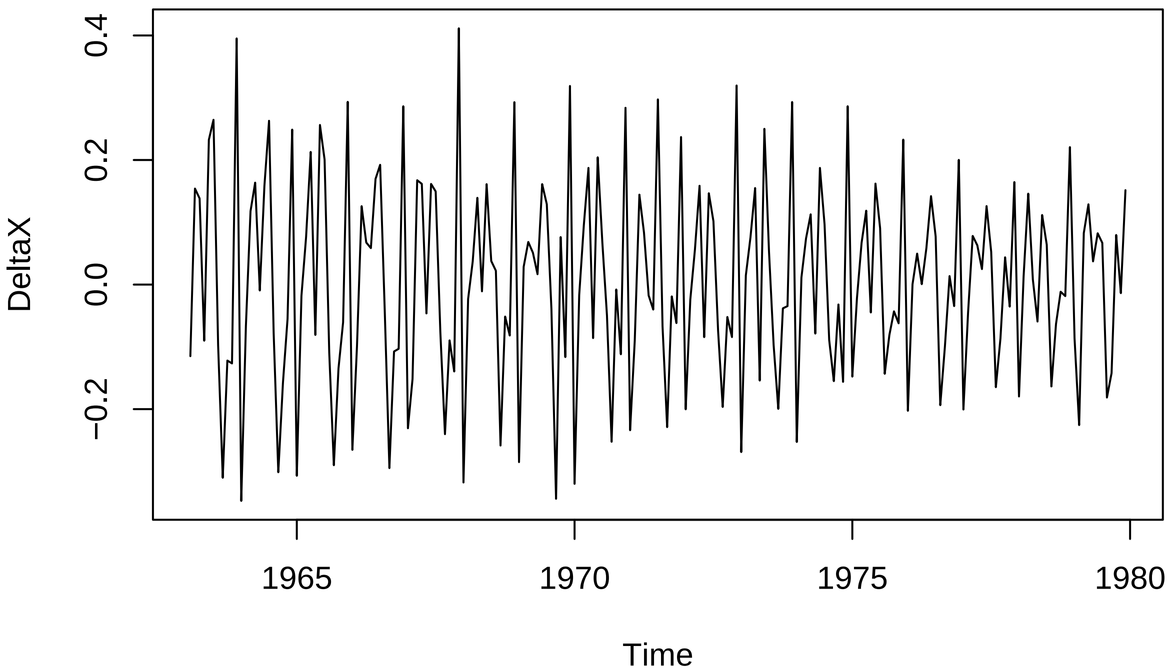
```
Box.test(X.train.log, lag = 20, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  X.train.log
## X-squared = 892.56, df = 20, p-value < 2.2e-16
```

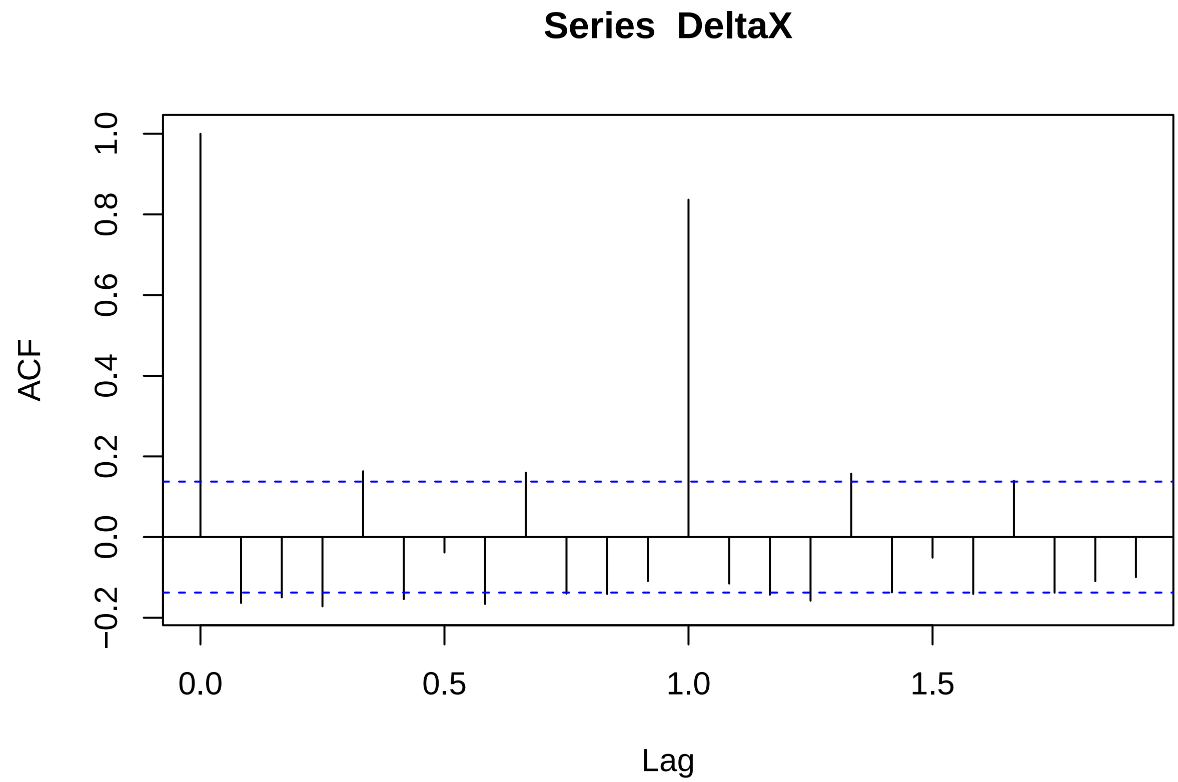
Avec une petite value p, donc, la probabilité que la série soit un bruit blanc est presque nulle

Nous commençons par différencier une fois.

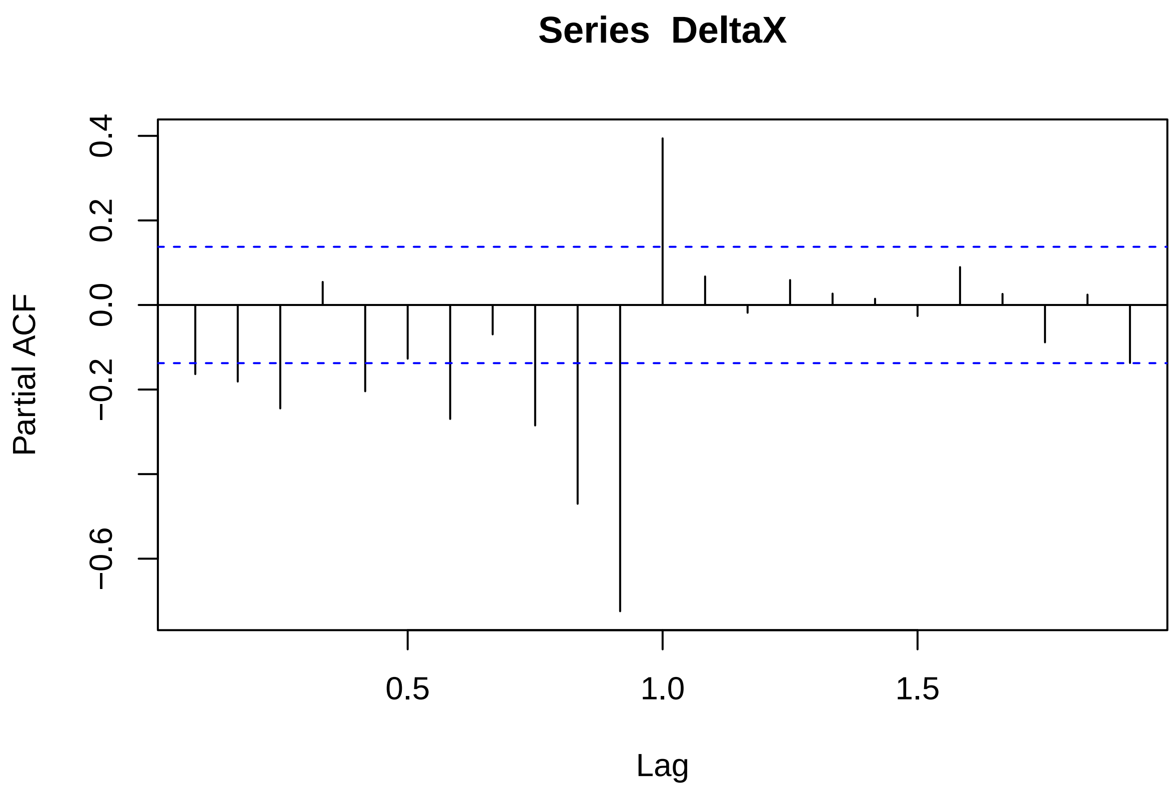
```
DeltaX=diff(X.train.log)
plot(DeltaX)
```




```
acf(DeltaX)
```




```
pacf(DeltaX)
```



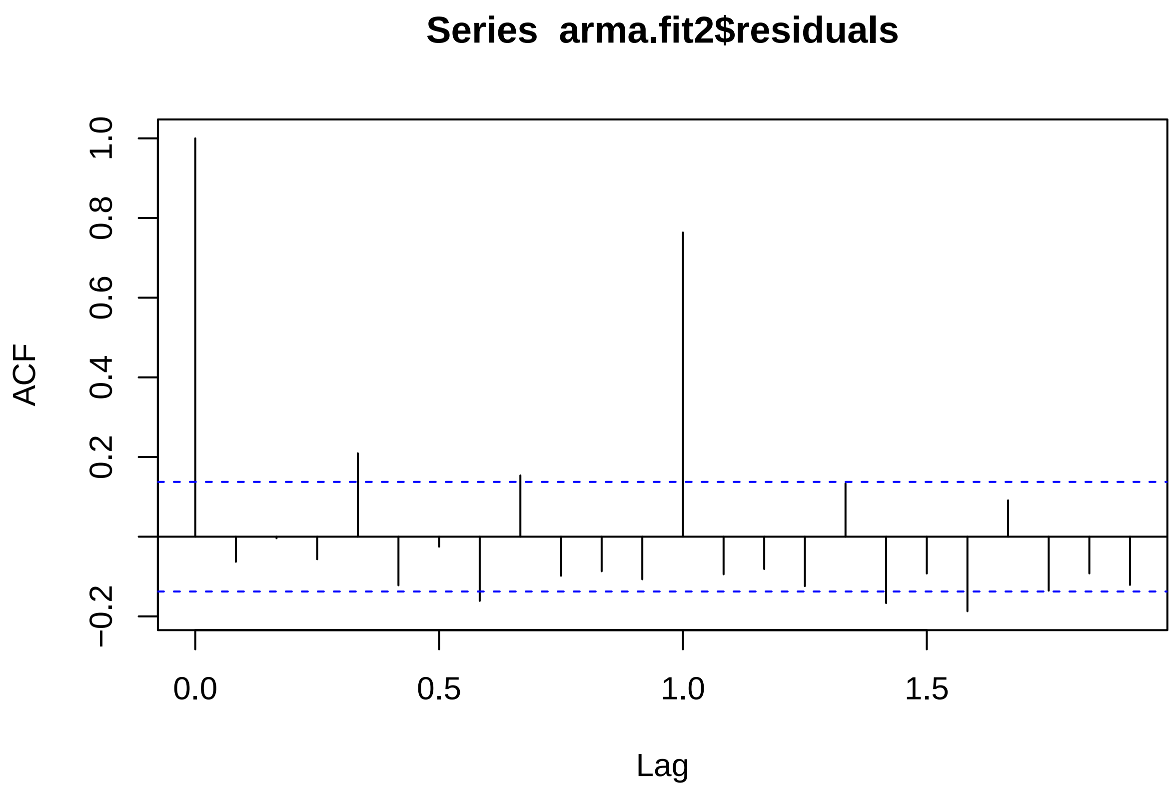
Les fonctions d'autocorrélation et d'autocorrélation partielles ressemblent à celles d'un ARMA(p,q).

Nous allons estimer un modèle ARMA(2,2)

```
arma.fit2 = arima(DeltaX,order=c(2,0,2))
print(arma.fit2)
```

```
##
## Call:
## arima(x = DeltaX, order = c(2, 0, 2))
##
## Coefficients:
##          ar1      ar2      ma1     ma2  intercept
##       1.3342  -0.6025  -1.9115  0.9359     0.0024
## s.e.  0.0586   0.0579   0.0395  0.0367     0.0008
##
## sigma^2 estimated as 0.01562:  log likelihood = 131.79,  aic = -251.57
```

```
acf(arma.fit2$residuals)
```



Les résidus sont bien améliorés mais cela réprésente tjrs une autocorrélation au lag 12. Donc, notre modèle a encore des choses à améliorer.

Modélisation

Approche 1 : comparer avec la fonction arma automatique

```
auto.arima(DeltaX,d=0,seasonal=FALSE)
```

```
## Series: DeltaX
## ARIMA(2,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1    mean
##        0.528  -0.1447  -0.9684  0.0025
## s.e.   0.071   0.0714   0.0204  0.0006
##
## sigma^2 = 0.01922:  log likelihood = 114.01
## AIC=-218.03   AICc=-217.72   BIC=-201.46
```

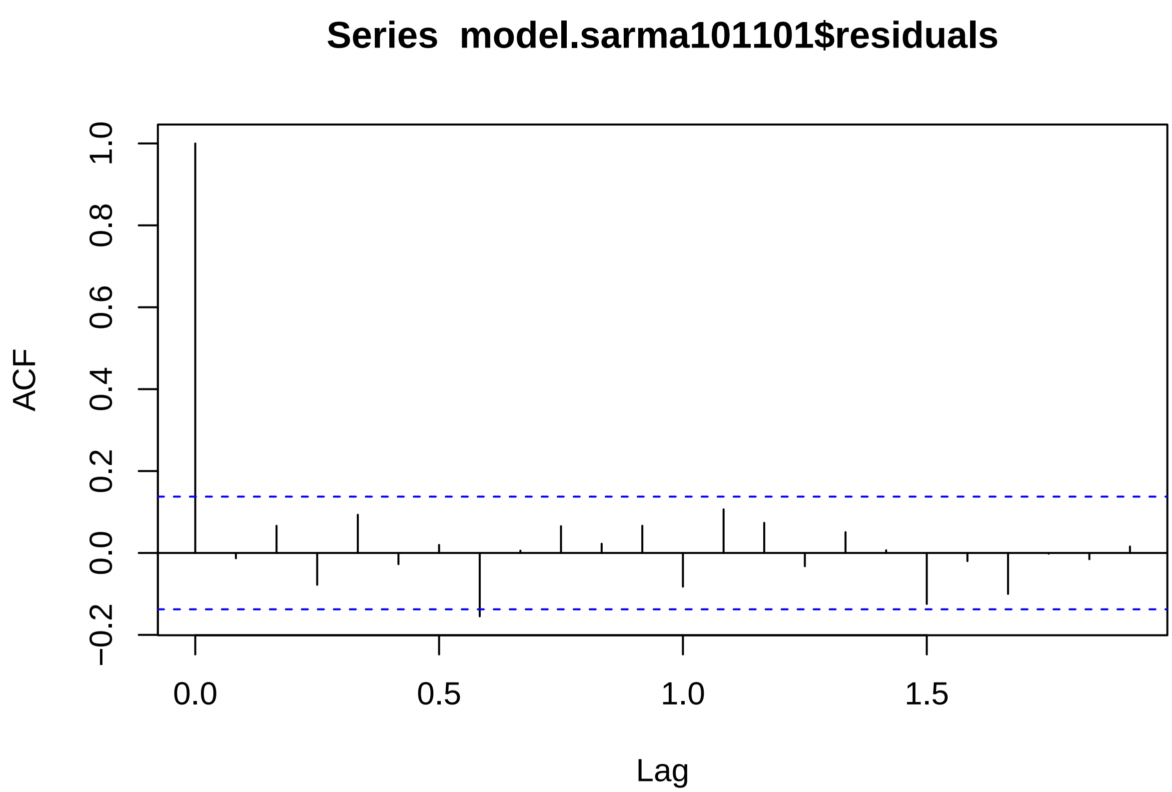
qui nous donne un ARMA(2,2) également. Mais en réalité, en regardant les résidus, ce n'est pas le bon modèle. Probalement, c'est à cause de la présentation de saisonalité des données.

Approche 2 : Dans ce cas-là, on va essayer de jouer avec SARMA (modèle ARMA saisonnier)

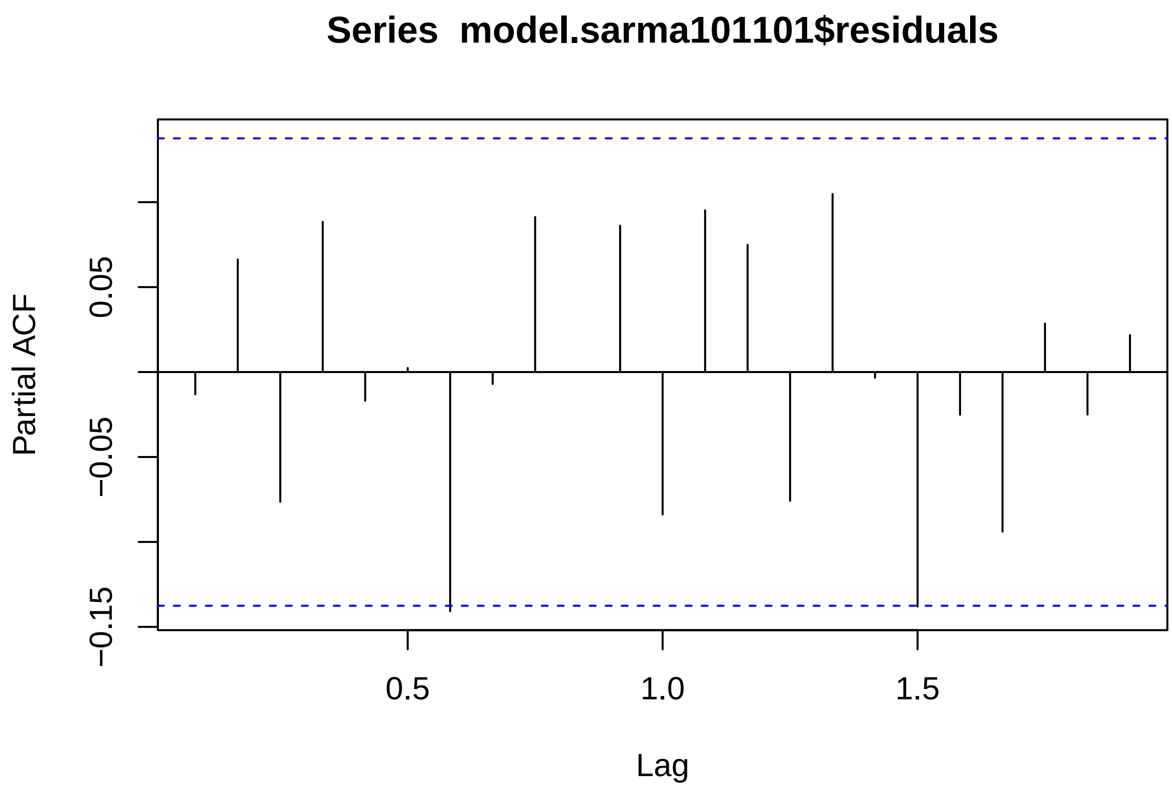
```
model.sarma101101 <- arima(DeltaX,order=c(1,0,1),seasonal=list(order=c(1,0,1),period=12))
model.sarma101101
```

```
##
## Call:
## arima(x = DeltaX, order = c(1, 0, 1), seasonal = list(order = c(1, 0, 1), period = 12))
##
## Coefficients:
##          ar1      ma1    sar1     sma1  intercept
##       0.1827  -0.9082  0.9843  -0.4491     0.0017
## s.e.  0.0815   0.0340  0.0071   0.0676     0.0047
##
## sigma^2 estimated as 0.002347:  log likelihood = 310.83,  aic = -609.65
```

```
acf(model.sarma101101$residuals)
```




```
pacf(model.sarma101101$residuals)
```



c'est bcp mieux que l'avant. On garde en tête que l'AIC de ce modèle est aic = -609.65 !! On essaie de comparer celui-ci avec celui de la fonction automatique

```
model.sarima.auto <- auto.arima(DeltaX,d=0,D=0)
model.sarima.auto
```

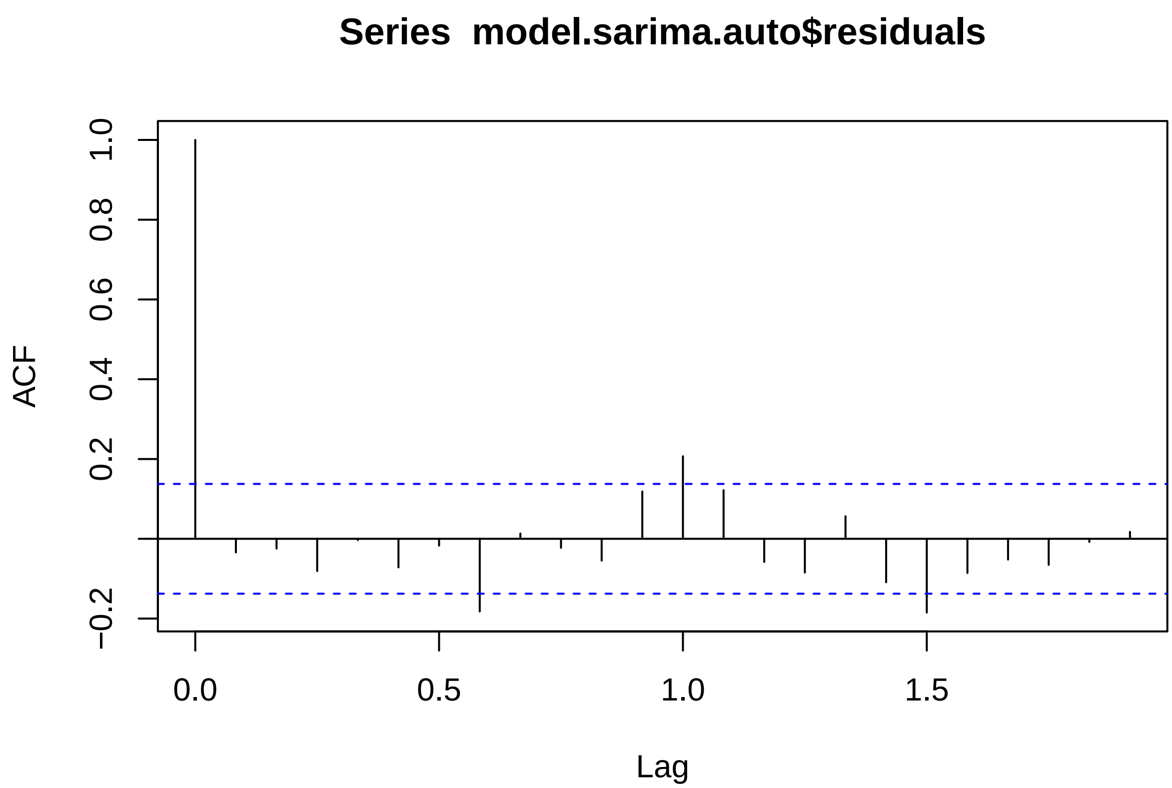
```
## Series: DeltaX
## ARIMA(0,0,3)(0,0,2)[12] with zero mean
##
## Coefficients:
##           ma1      ma2      ma3    sma1    sma2
##       -0.5385  -0.2242  -0.1476  0.8337  0.6612
## s.e.   0.0743   0.0752   0.0674  0.0639  0.0664
##
## sigma^2 = 0.006056:  log likelihood = 223.86
## AIC=-435.72   AICc=-435.29   BIC=-415.84
```

```
model.sarima.auto.bestAic = auto.arima(DeltaX,d=0,D=0,ic="aic")
model.sarima.auto.bestAic
```

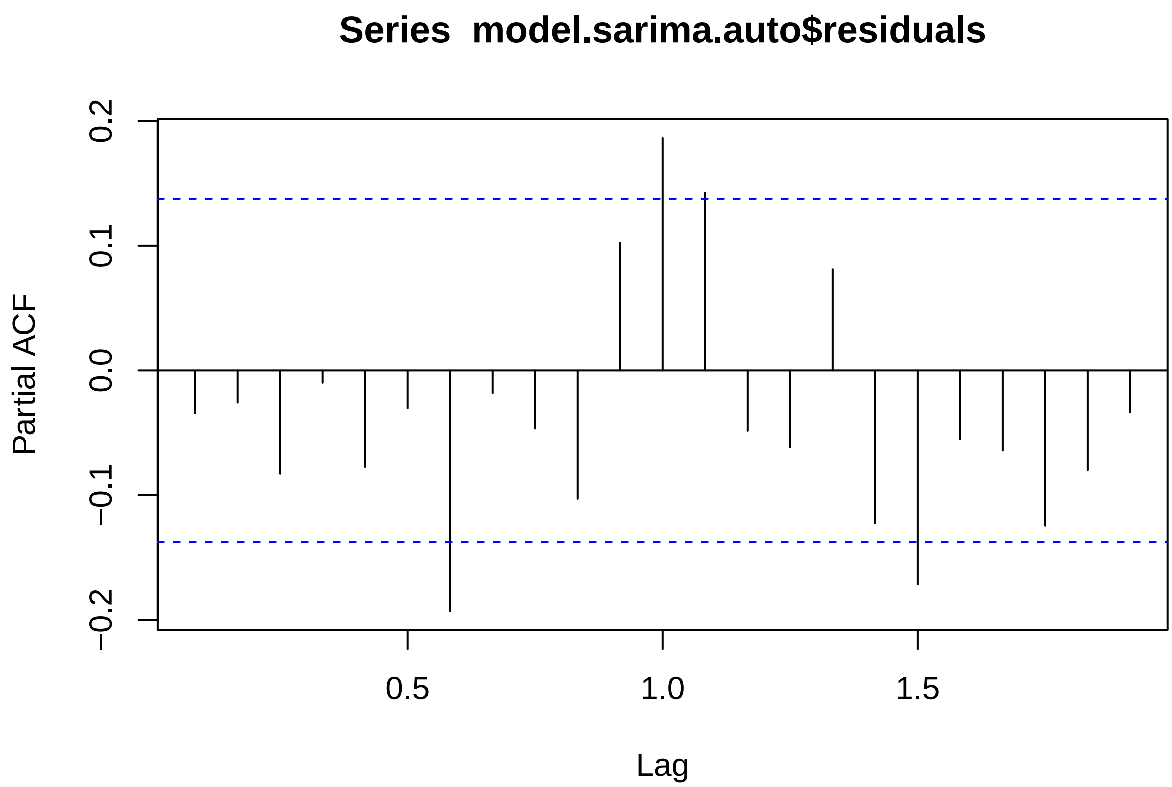
```
## Series: DeltaX
## ARIMA(1,0,3)(0,0,2)[12] with non-zero mean
##
## Coefficients:
##           ar1     ma1      ma2      ma3    sma1    sma2    mean
##       -0.7723  0.2005  -0.7457  -0.3853  0.8021  0.6659  0.0024
```

```
## s.e.   0.1804  0.1812   0.1334   0.0724  0.0633  0.0677  0.0006
##
## sigma^2 = 0.005852:  log likelihood = 228.11
## AIC=-440.21   AICc=-439.47   BIC=-413.71
```

```
acf(model.sarima.auto$residuals)
```




```
pacf(model.sarima.auto$residuals)
```

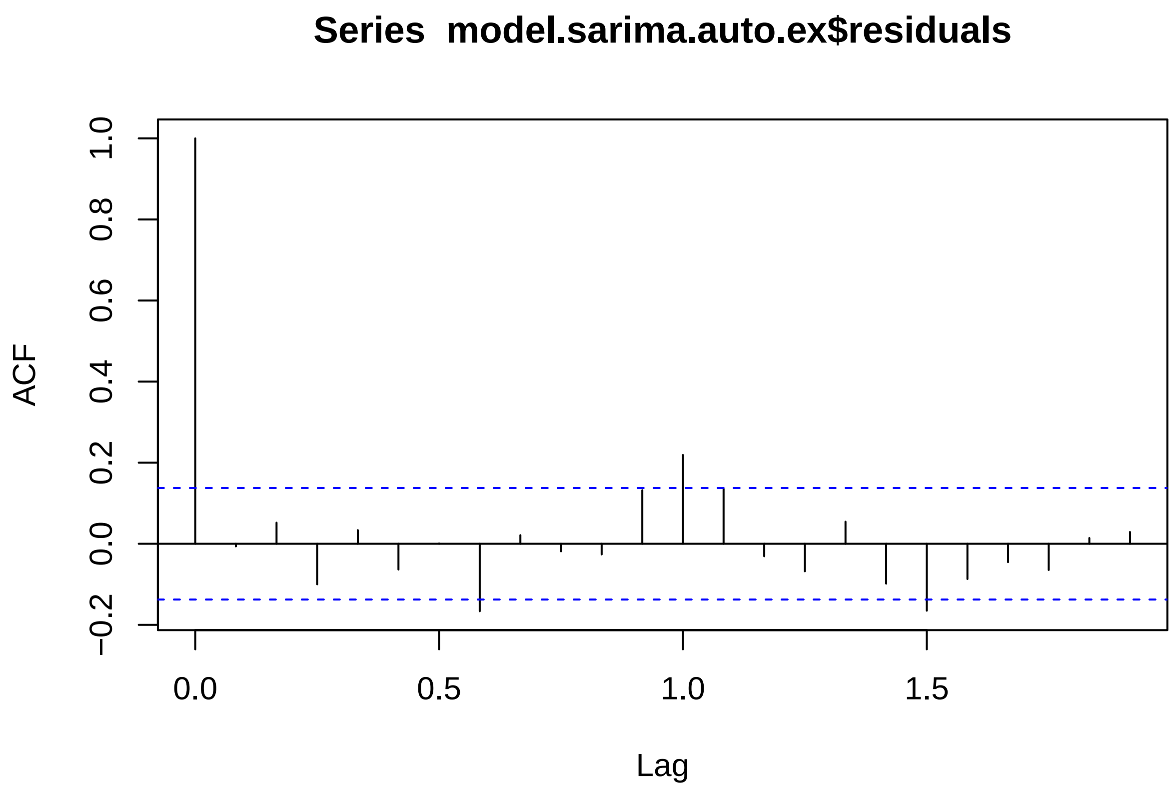


On a obtenu un autre modèle avec un AIC à -435.72 c'est un grand écart à model.sarma101101 et le graph PACF n'est pas trop idéal. La fonction n'a donc pas permis de détecter le meilleur modèle. On va essayer de faire une recherche exhautive

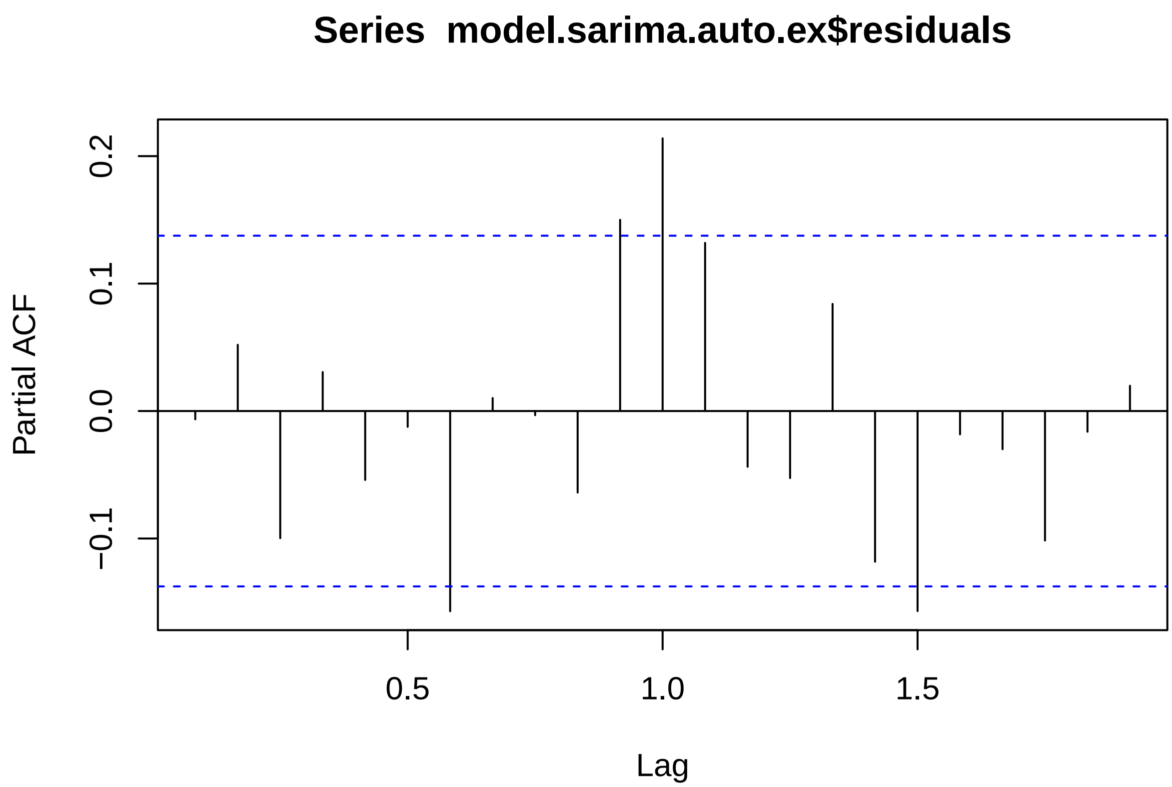
```
model.sarima.auto.ex <- auto.arima(DeltaX,d=0,D=0,stepwise=FALSE,approximation=FALSE)
model.sarima.auto.ex
```

```
## Series: DeltaX
## ARIMA(2,0,1)(0,0,2)[12] with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1    sma1    sma2    mean
##       0.3974  -0.0551  -0.9783  0.8310  0.6606  0.0024
## s.e.  0.0744   0.0781   0.0245  0.0653  0.0667  0.0005
##
## sigma^2 = 0.005934:  log likelihood = 226.25
## AIC=-438.51   AICc=-437.93   BIC=-415.32
```

```
acf(model.sarima.auto.ex$residuals)
```




```
pacf(model.sarima.auto.ex$residuals)
```



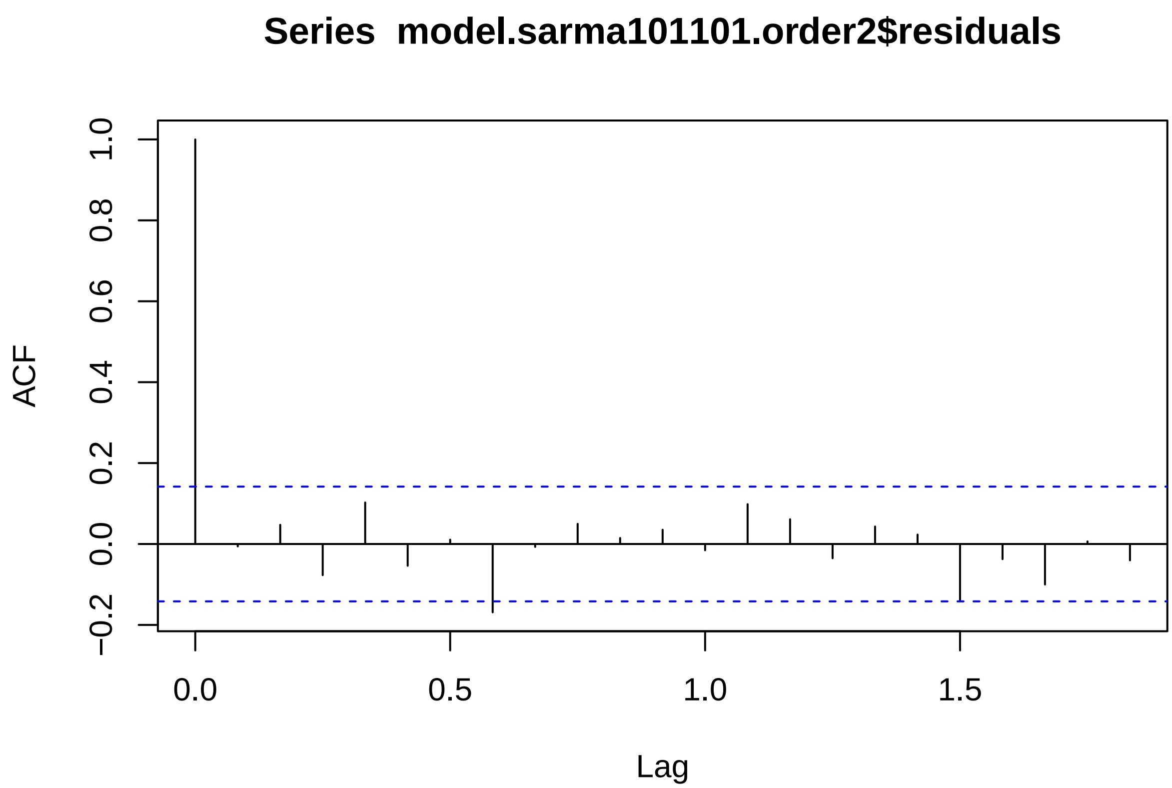
on voit bien encore un pic au lag 12. On continue notre modélisation en faire 2 fois différenciation. Un modèle sans différenciation suppose que la série originale est stationnaire. Un modèle avec une différenciation d'ordre 1 suppose que la série originale présente une tendance constante. Un modèle avec une différenciation d'ordre 2 suppose que la série originale présente une tendance variant dans le temps

Approche 3 : Modélisation d'un processus de type SARMA à l'aide de différenciation à 2 fois

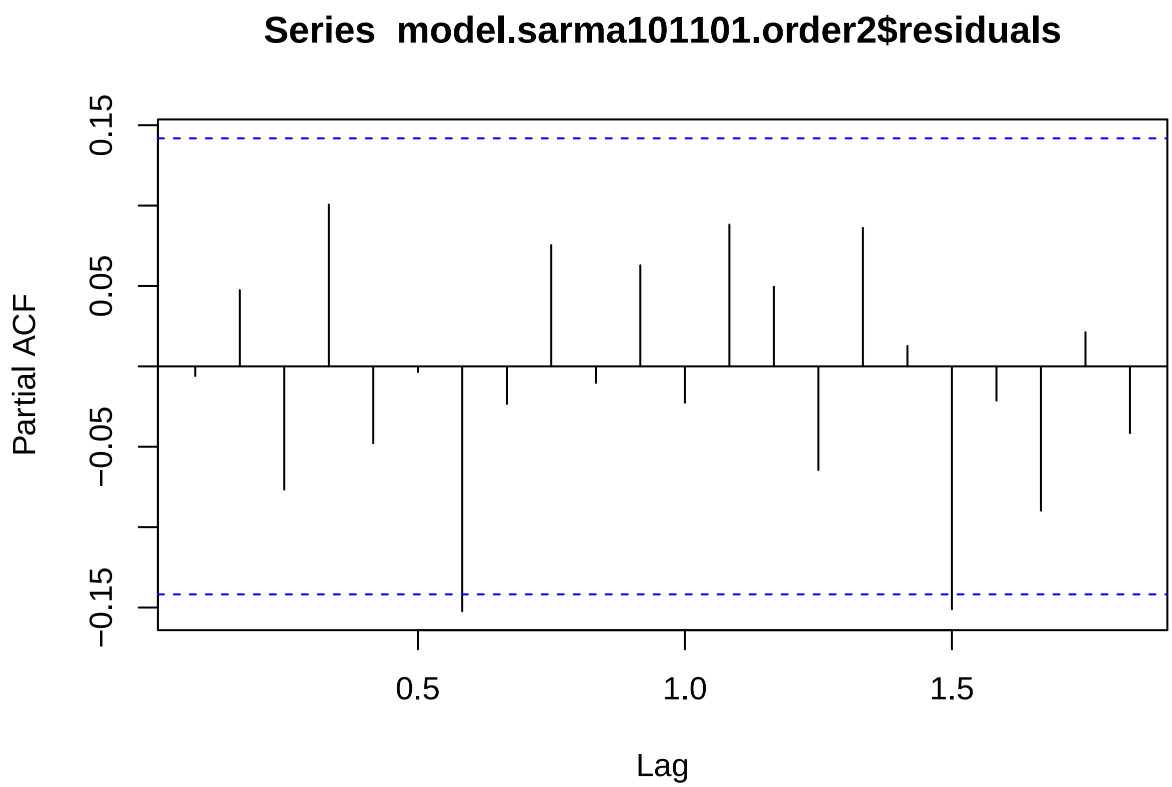
```
Delta12DeltaX=diff(DeltaX,lag=12)
model.sarma101101.order2 = arima(Delta12DeltaX,order=c(1,0,1),seasonal=list(order=c(1,0,1),period=12))
model.sarma101101.order2
```

```
##
## Call:
## arima(x = Delta12DeltaX, order = c(1, 0, 1), seasonal = list(order = c(1, 0,
##     1), period = 12))
##
## Coefficients:
##          ar1      ma1     sar1     sma1  intercept
##       0.1362  -0.8897  -0.1169  -0.3678      1e-04
## s.e.  0.0860   0.0417   0.1507   0.1379      3e-04
##
## sigma^2 estimated as 0.00235:  log likelihood = 304.82,  aic = -597.64
```

```
acf(model.sarma101101.order2$residuals)
```



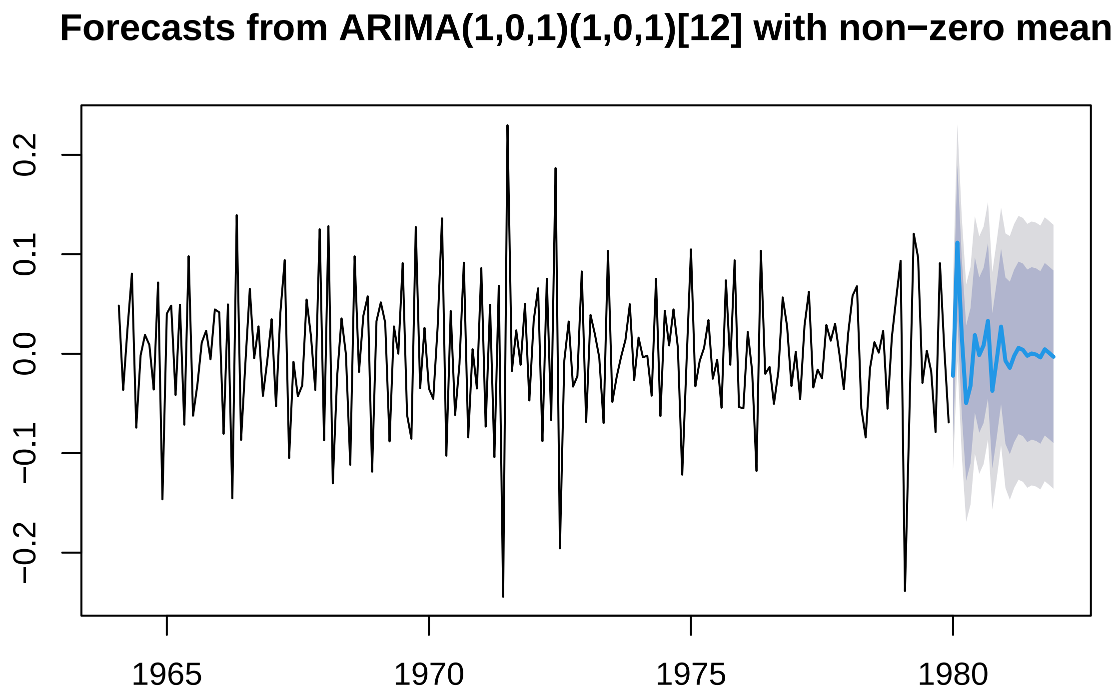

```
pacf(model.sarma101101.order2$residuals)
```



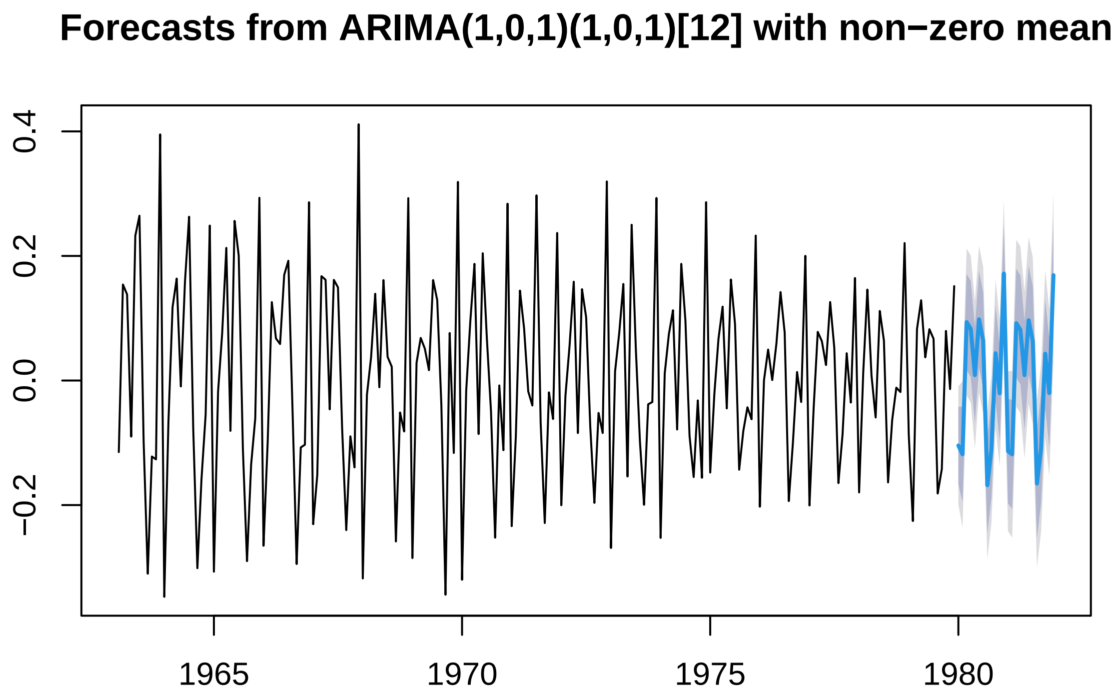
Voilà, le graphe nous convient mais les graphs ACF et PACF sont moins bien que celui de model.sarma101101. Le modèle nous donne un AIC à -597.64. On va comparer le résultat de la prédicition.

Prévision

```
model.sarma101101.order2.predict=forecast(model.sarma101101.order2 )
plot(model.sarma101101.order2.predict)
points(X.test.log,lwd=2,col="darkgreen",type='l')
```




```
model.sarma101101=forecast(model.sarma101101 )
plot(model.sarma101101)
points(X.test.log,lwd=2,col="darkgreen",type='l')
```

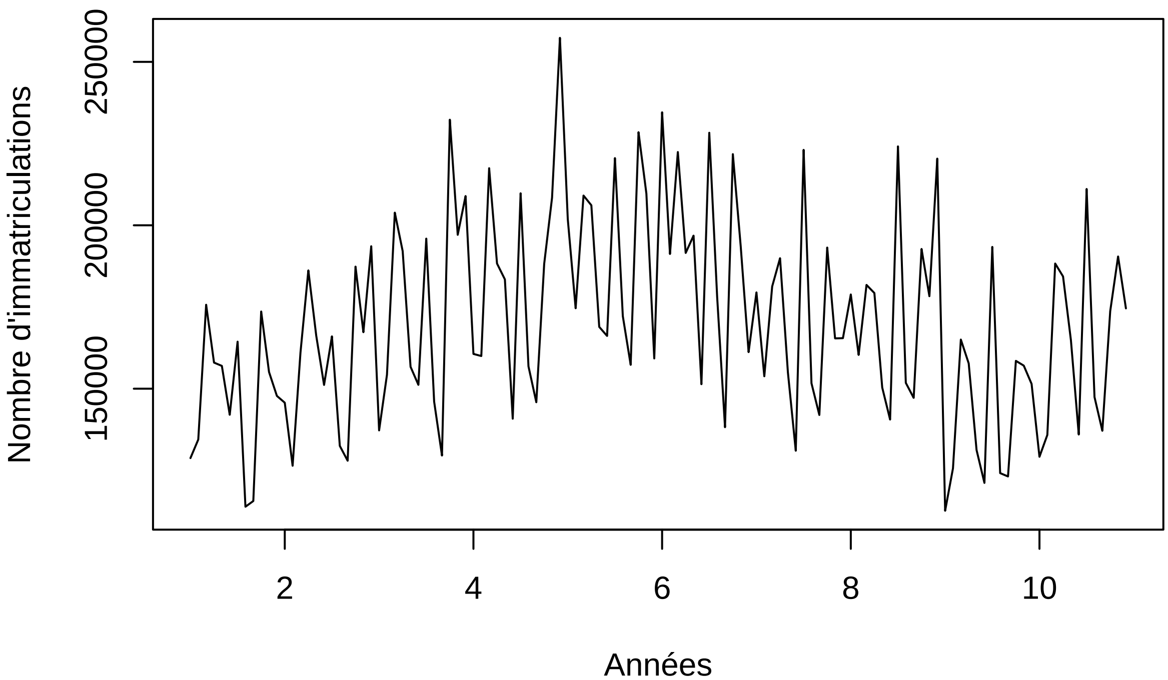


Visuellement, on voit que model.sarma101101 a mieux prédit.

II : Numéro d'Immatricualation

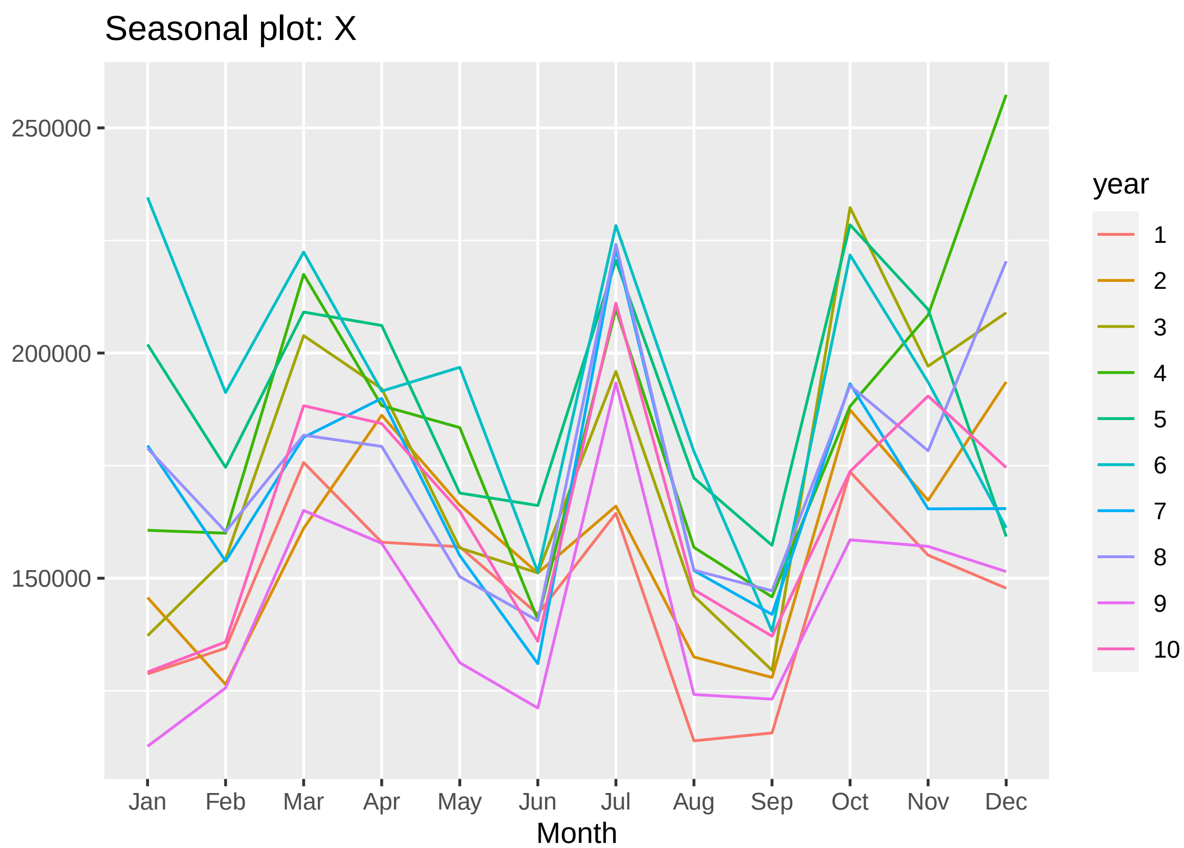
Chargement et visualisation des données

```
library(readxl)
immat <- read_excel("c7ex2.xls")
X <- ts(immat[!is.na(immat[,2]),2],frequency = 12)
plot(X,ylab="Nombre d'immatriculations",xlab="Années")
```

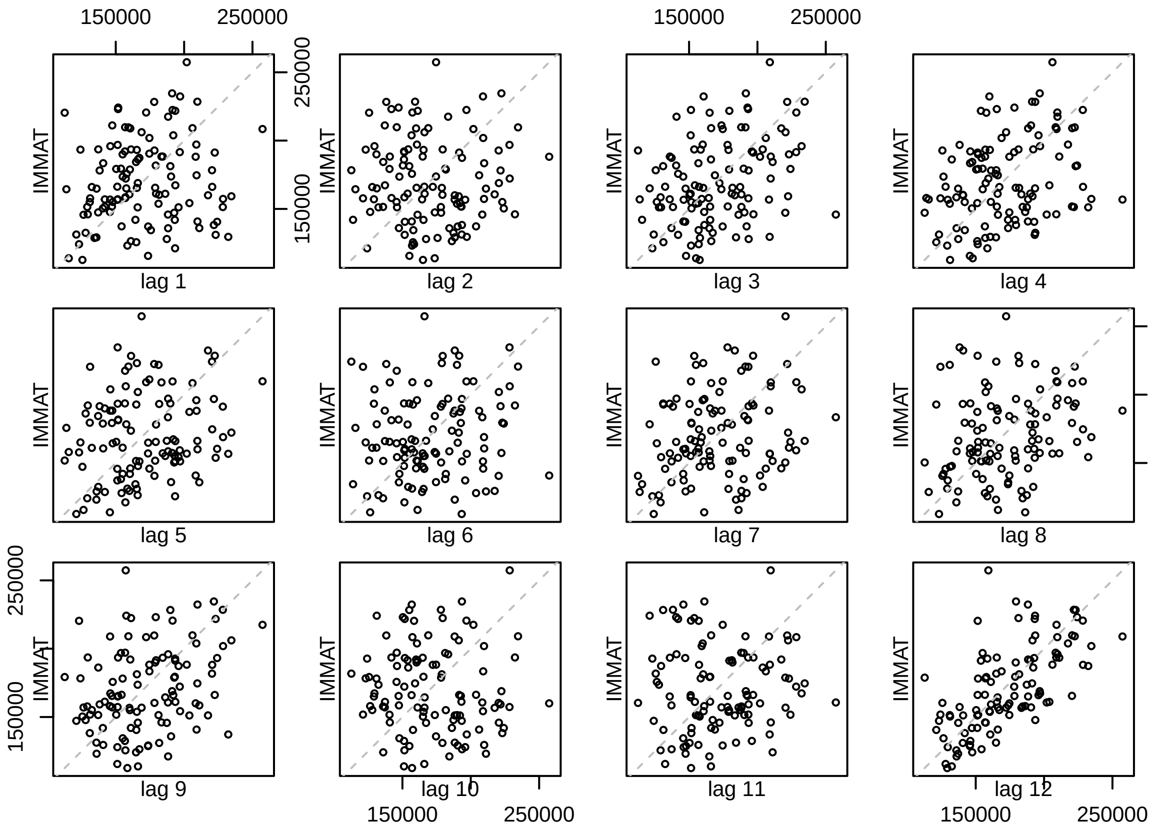


Saisonalité, tendance et résidus.

```
forecast::ggseasonplot(X)
```




```
lag.plot(X,lags=12,layout=c(3,4),do.lines=FALSE)
```



On voit bien qu'il y a une saisonalité marquée sur l'année mais pas claire pour le mois décembre et le mois Janvier et une forte corrélation au lag 12, ce qui nous fait penser à une série saisonnière.

Tests de stationnarité

On va faire un test de KPSS Unit Root Test

```
## On prends les données d'entraînement et les données de test pour évaluer notre qualité de prédicitio
X.train = window(X,end=c(9,12))
X.test <- window(X,start=10)

## On cherche à stabiliser les variances en faisant la transformation logarithme
X.train.log = log(X.train)
X.test.log = log(X.test)

## KPSS Unit Root Test
library(urca)
testKPSStau <- ur.kpss(X.train.log,type='tau')
summary(testKPSStau)
```

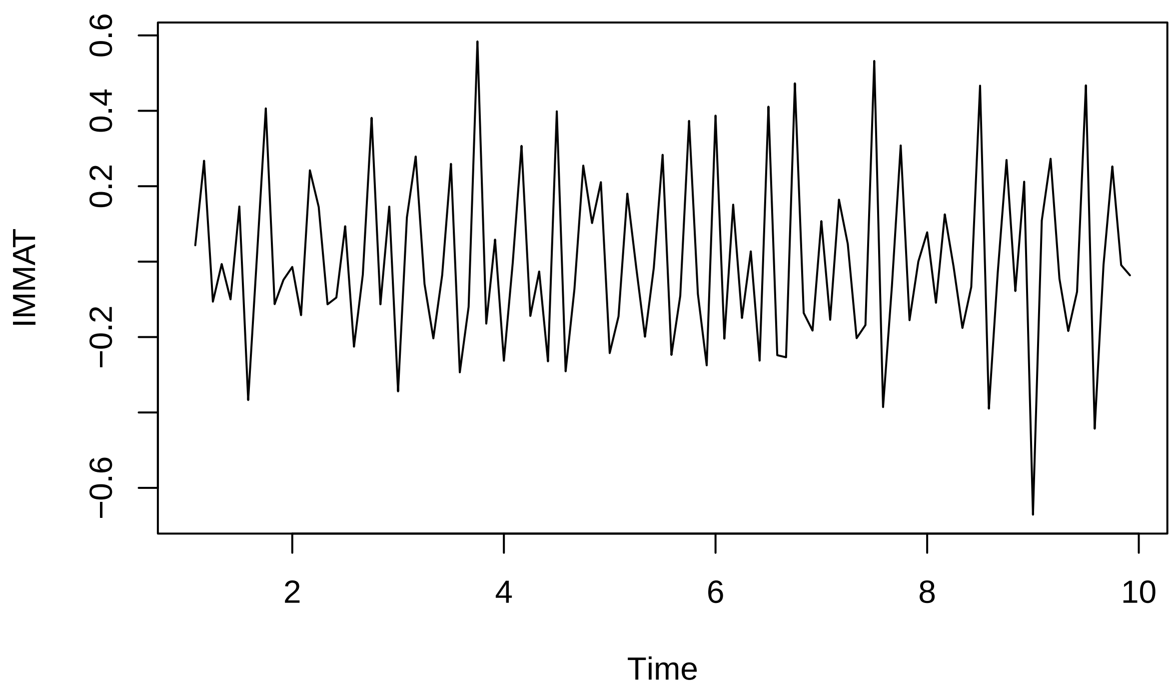
```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: tau with 4 lags.
```

```
##
## Value of test-statistic is: 0.4428
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

Vu que la p-value est important > 0.05, donc, notre jeu de données n'est pas une série stationnaire. Il y a une marche aléatoire dans notre série. On va faire la différenciation une fois pour éliminer la marché aléatoire

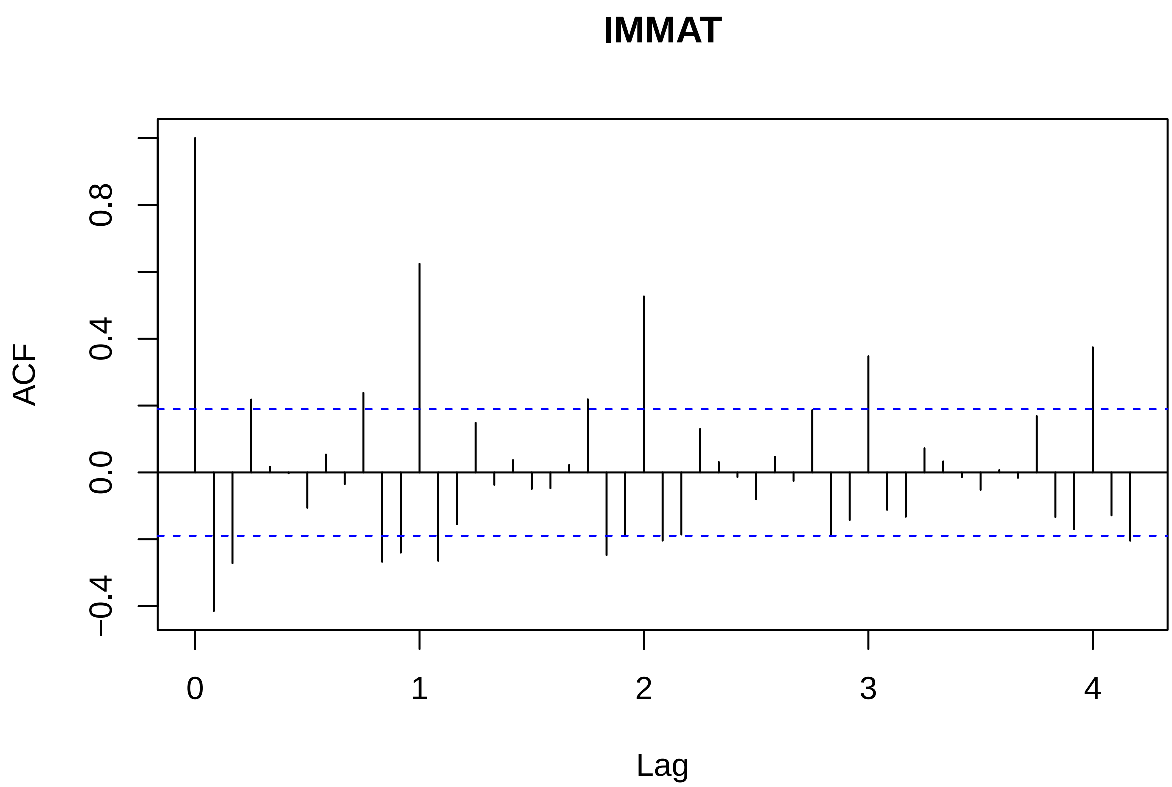
Différenciation

```
X.train.log.delta1 =diff(X.train.log)
plot(X.train.log.delta1)
```

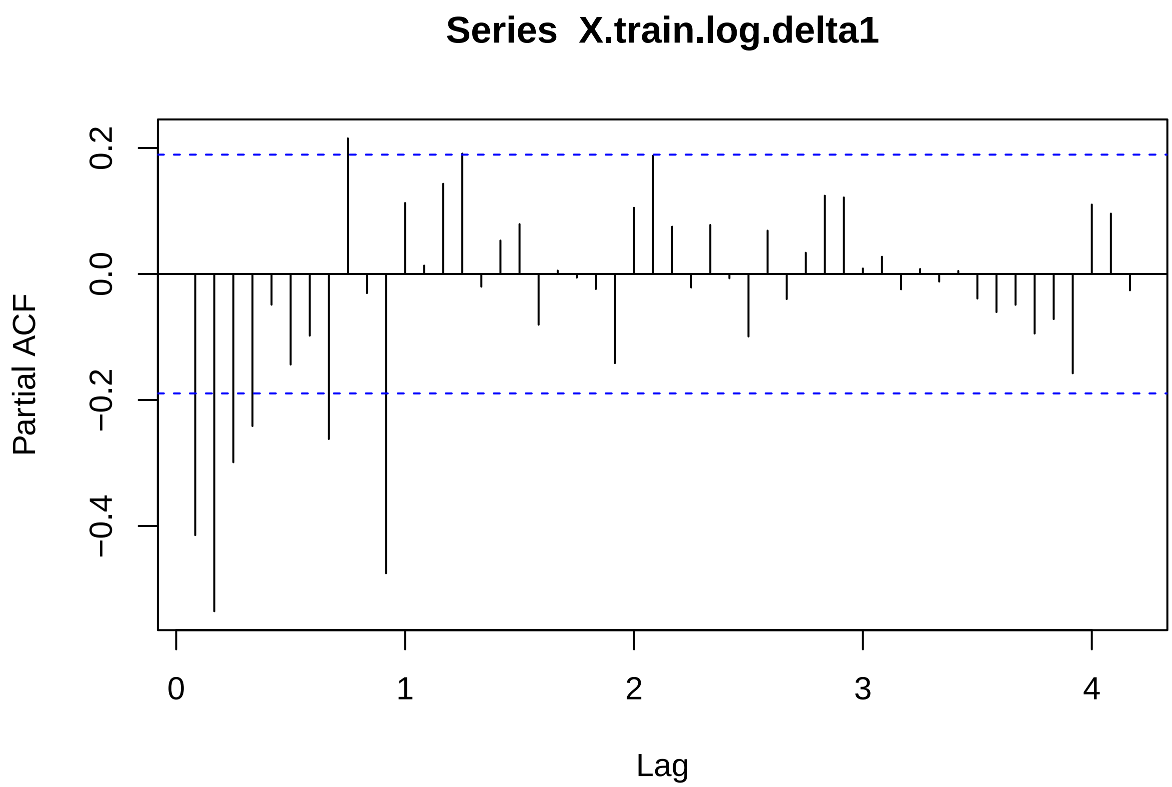


De ce qu'on voit, il n'y a pas clairement de la tendance dans la série transformée. On va regarder l'autocorélation

```
acf(X.train.log.delta1,lag.max=50)
```

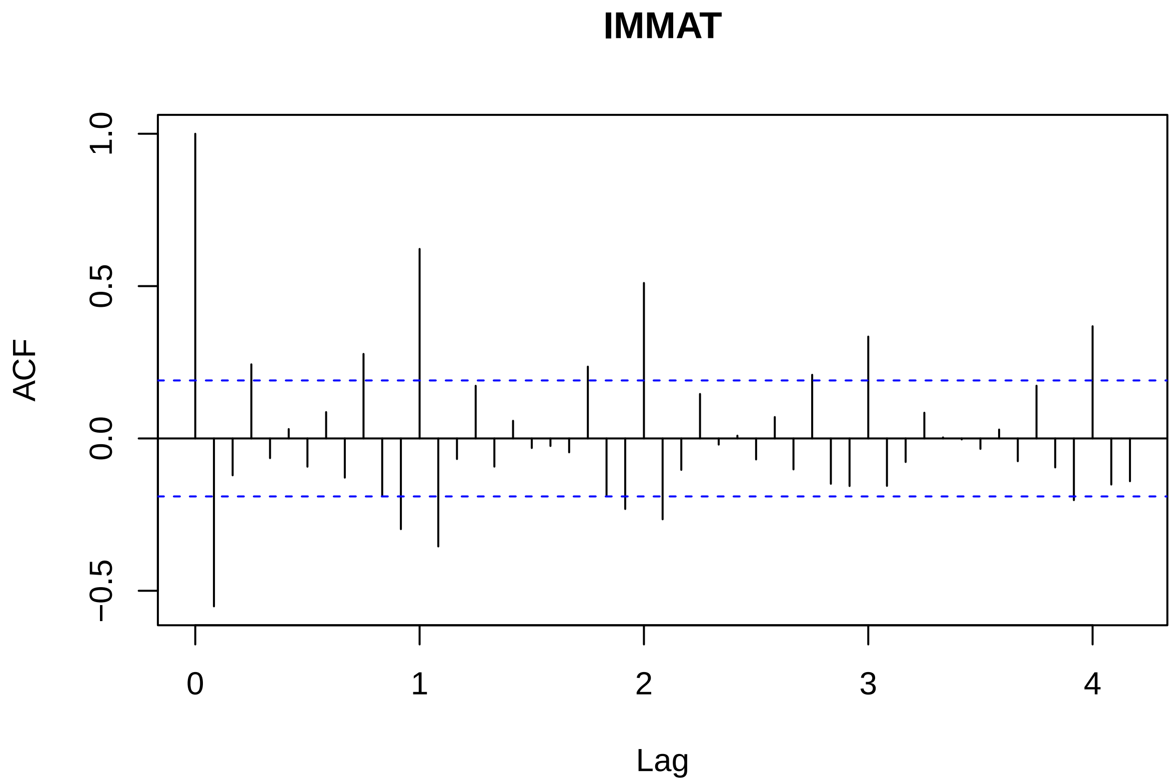



```
pacf(X.train.log.delta1,lag.max=50)
```

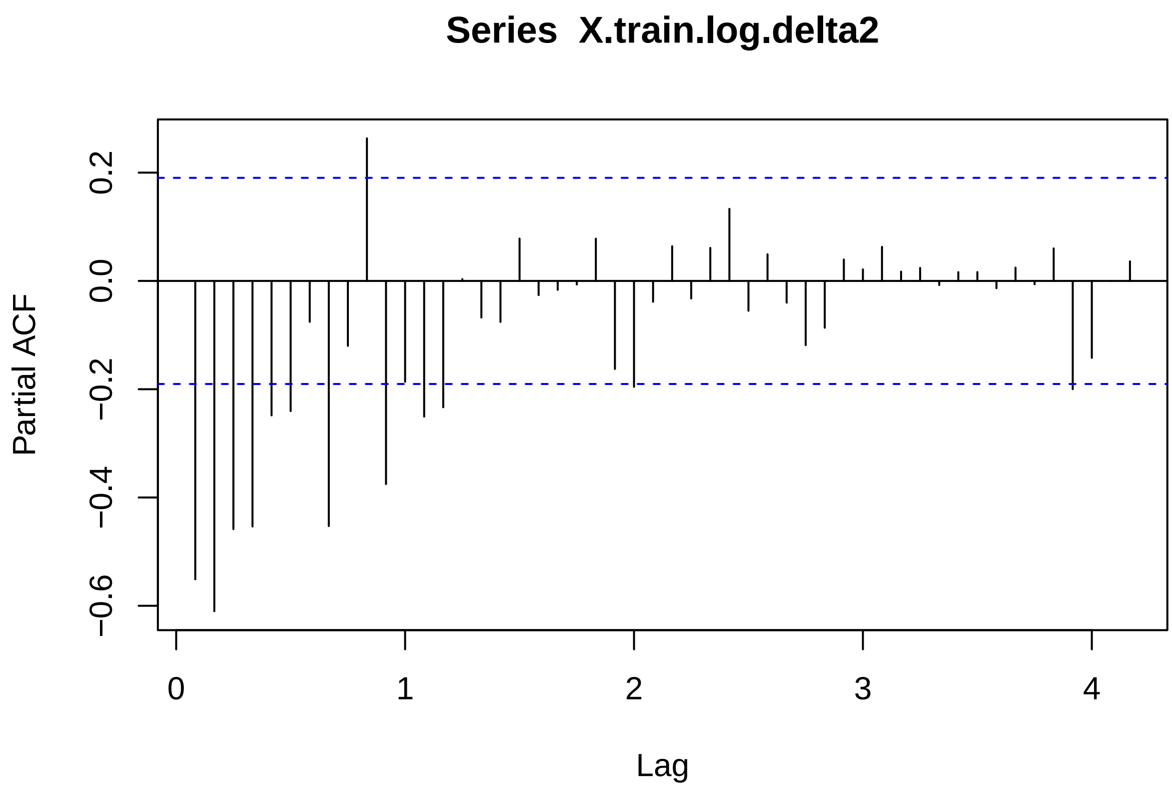


il y a un pic au lag 1 au ACF et le pacf représent une décroissance. Vu qu'il y a une saisonalité, on pourrait penser à un modèle SARIMA(0,1,q)(0,D,Q). Nous allons faire une différentiation saisonnière pour supprimer la saisonalité.

```
 X.train.log.delta2 =diff(diff(X.train.log))
acf( X.train.log.delta2,lag.max=50)
```



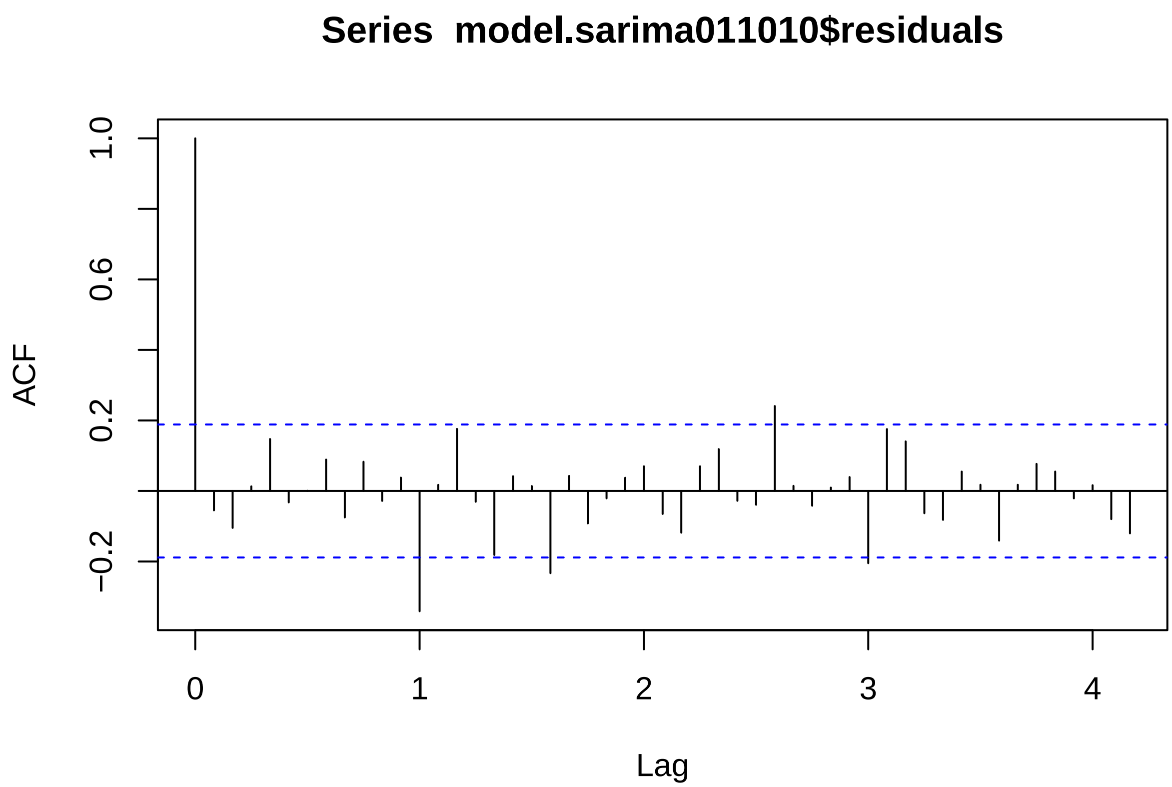

```
pacf( X.train.log.delta2,lag.max=50)
```



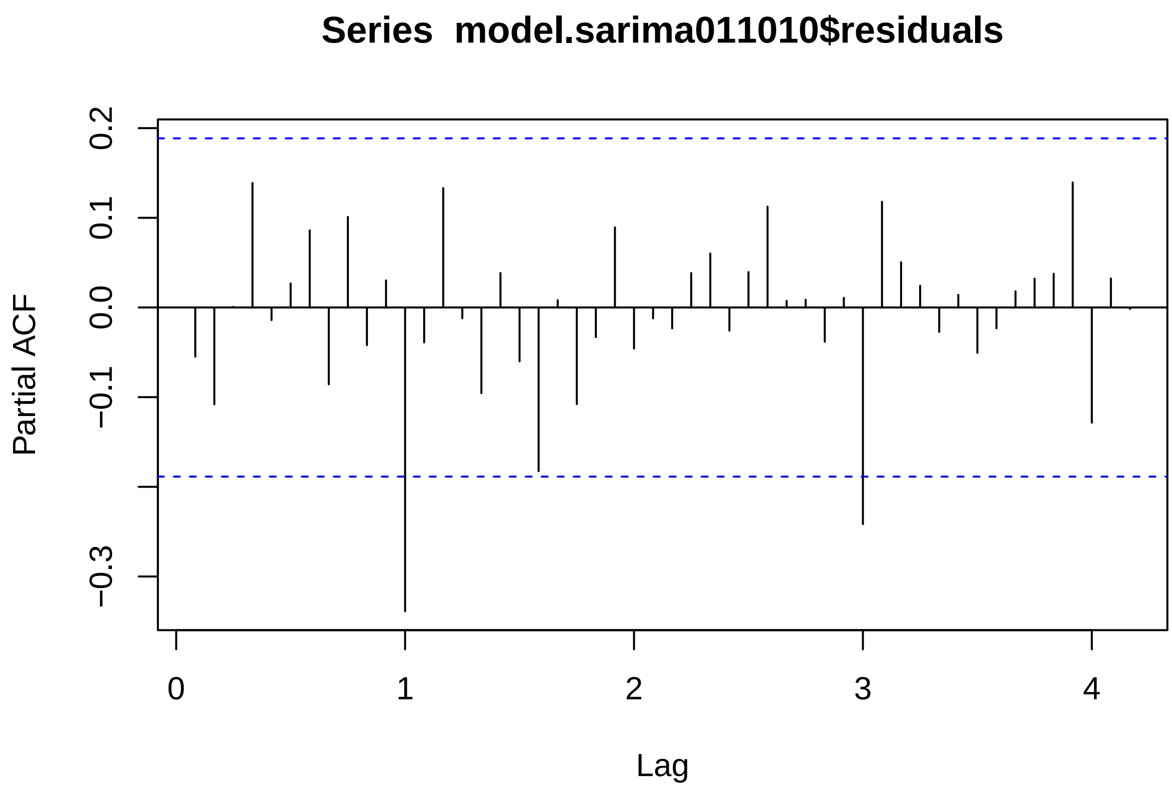
La acf représente un pic au retard 1 et on voit aussi une décroissance exponentielle de la pacf.Selon le cours,on peut essayer avec le modèle SARIMA(0, 1, 1)(0, 1, 0)[12]

Modélisation

```
model.sarima011010 <- arima(X.train.log,order=c(0,1,1),seasonal=list(order=c(0,1,0),period=12))
acf(model.sarima011010$residuals,lag.max=50)
```




```
pacf(model.sarima011010$residuals,lag.max=50)
```

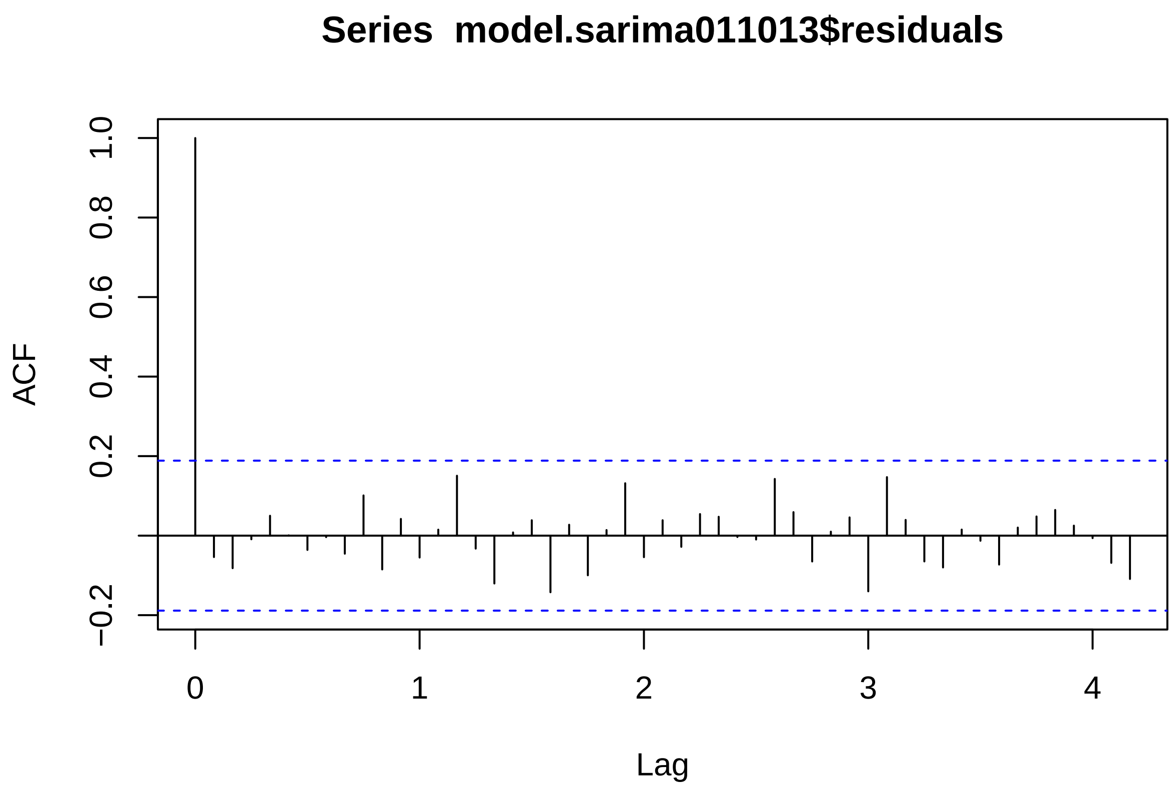


Les graphs nous montrent encore des pics non significatif. ce dernier nous explique ce modèle n'est pas bien modélisé. On cherche à changer les ordres P,Q de saisonalité dans le modèle pour obtenir le meilleur modèle.

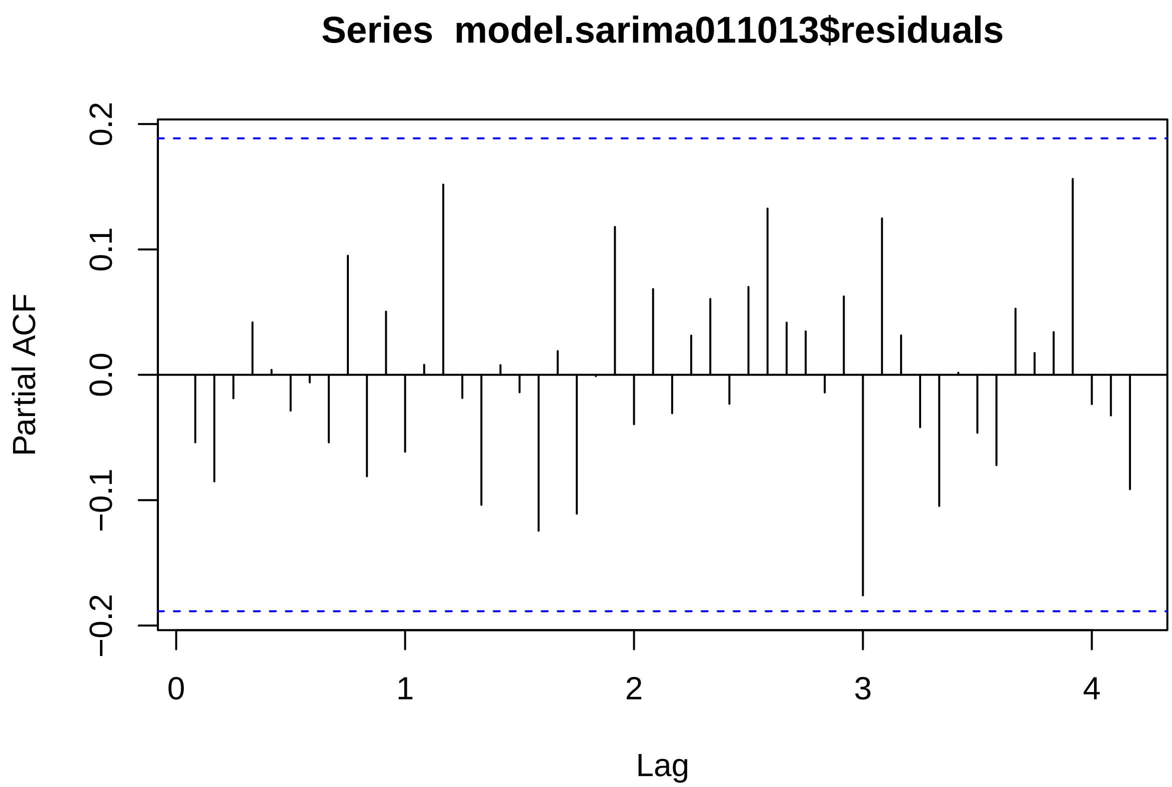
```
model.sarima011013 <- arima(X.train.log,order=c(0,1,1),seasonal=list(order=c(0,1,3),period=12))
summary(model.sarima011013)
```

```
##
## Call:
## arima(x = X.train.log, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 3),
##     period = 12))
##
## Coefficients:
##           ma1     sma1     sma2     sma3
##       -0.7994  -0.6530  -0.0170  -0.3299
## s.e.   0.0575   0.2515   0.1437   0.1467
##
## sigma^2 estimated as 0.01122:  log likelihood = 68.45,  aic = -126.9
##
## Training set error measures:
##                       ME       RMSE        MAE        MPE      MAPE     MASE
## Training set -0.01441088 0.09942436 0.07049681 -0.1249002 0.5859839 0.374361
##                     ACF1
## Training set -0.05387697
```

```r
acf(model.sarima011013$residuals,lag.max=50)
```




```r
pacf(model.sarima011013$residuals,lag.max=50)
```



les graphs sont maintenant bien améliorés et ce modèle nous donne un AIC à -126.9 On va continuer à comparer avec celui proposé par la fonction automatique

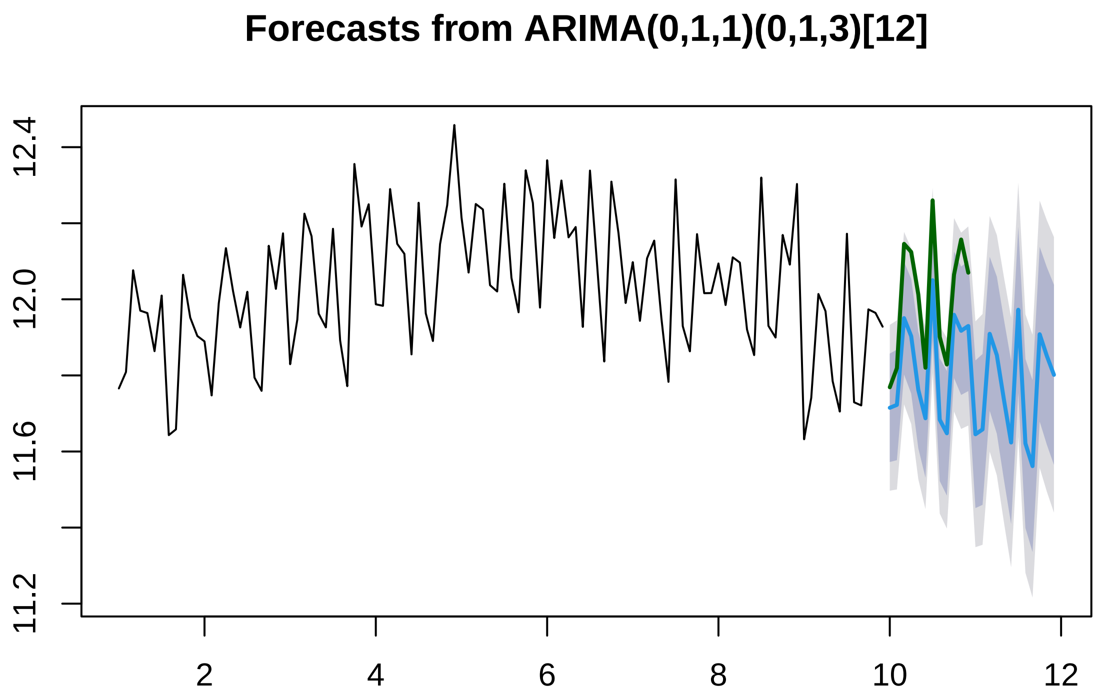
```
model.sarima.auto <- auto.arima(X.train.log)
summary(model.sarima.auto)
```

```
## Series: X.train.log
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##           ma1     sma1
##       -0.8058  -0.5998
## s.e.   0.0564   0.1380
##
## sigma^2 = 0.01403:  log likelihood = 65.7
## AIC=-125.4   AICc=-125.14   BIC=-117.74
##
## Training set error measures:
##                       ME      RMSE        MAE        MPE      MAPE      MASE
## Training set -0.01557684 0.1099079 0.07333541 -0.1351425 0.6094104 0.6402572
##                     ACF1
## Training set -0.0665462
```

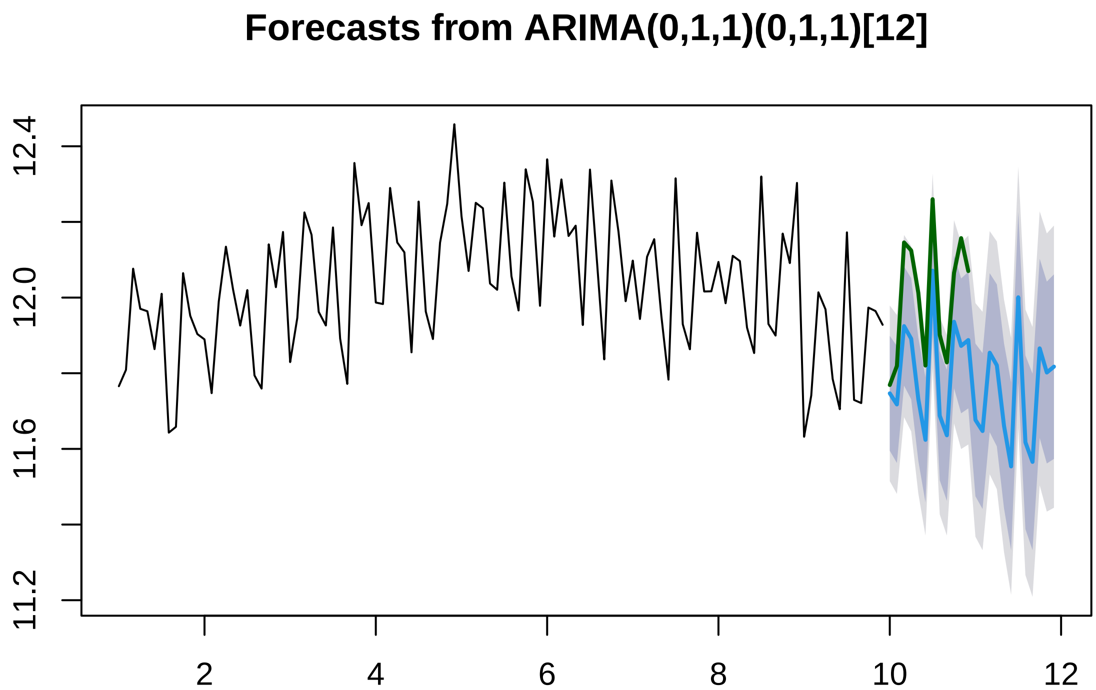
On a obtenu un SARIMA(0,1,1)(0,1,1)[12] avec un AIC à -125,4. Pour évaluer lequel serait meilleur, on va évaluer la qualité de prédiction

Prédiction

```
model.sarima011013.pred = forecast(model.sarima011013 )
plot(model.sarima011013.pred)
points(X.test.log,lwd=2,col="darkgreen",type='l')
```




```
model.sarima.auto.pred = forecast(model.sarima.auto )
plot(model.sarima.auto.pred)
points(X.test.log,lwd=2,col="darkgreen",type='l')
```



Visuellement, on voit bien que le modèle (0,1,1)(0,1,3) a mieux prédit que celui de (0,1,1)(0,1,1)