Projet séries Temporelles

SEFFANE Asmaa

EMSBD

2023-06-05

Je m'intéresse dans ce projet à étudier les séries temporelles. Il y aura deux jeux de données,

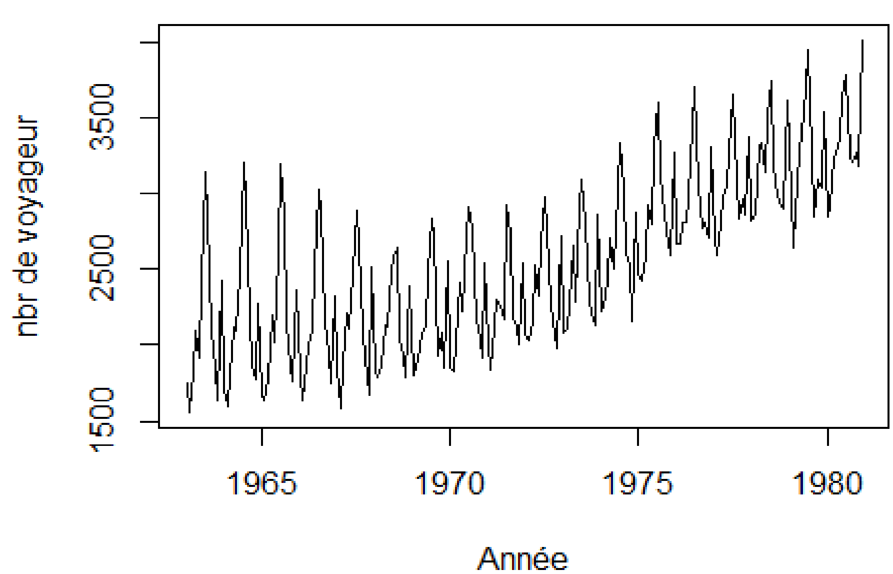
- un sur les nombre des voyageurs sur le réseau SNCF,
- et un autre sur le nombre des immatriculations de voitures particulières en France.

Jeu de données SNCF

J'étudie dans ce jeu de données le nombre de voyageurs sur le réseau SNCF.

Je commence par télécharger les Library nécessaire et mes données :

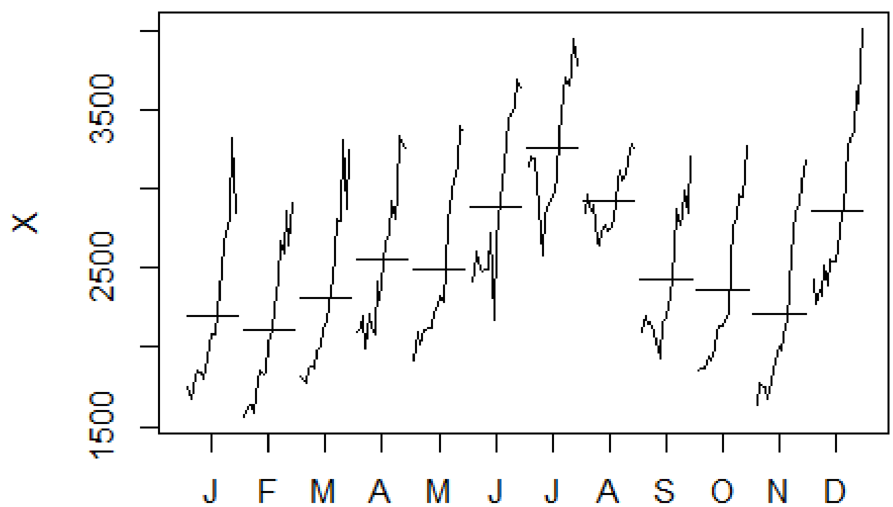
```
library(forecast)

library(caschrono)

library(stats)

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)
plot(X, xlab = "Année", ylab = "nbr de voyageur")
```



La série croit au cours du temps. Il n'y a pas de linéarité. Il y a aussi une saisonnalité

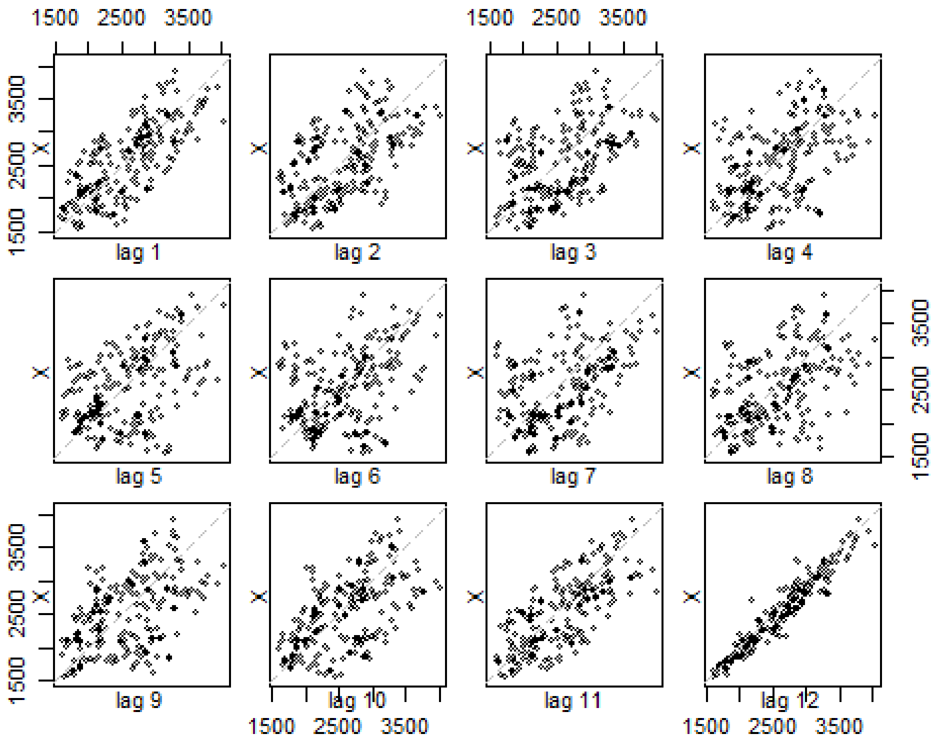
Je notice aussi une variance qui n'est pas constante et qui diminue avec le temps.

Le chronogramme par mois



Il y a une augmentation maximale pendant l'été et en décembre aussi ce qui coïncide avec les vacances. La saisonnalité est bien marquée.

Le Lag out :



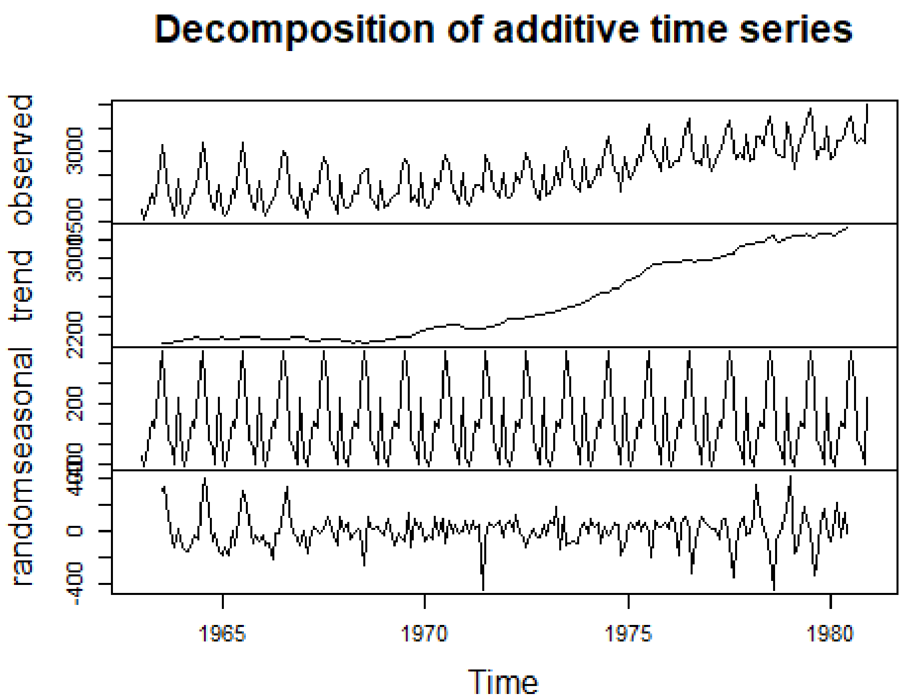
La série semble avoir à la fois une tendance et une composante saisonnière.

La série décalé de 12 mois est bien corrélée et la corrélation linéaire est bien claire.

J'utilise La fonction 'decompose' de R pour aider à modéliser notre série.

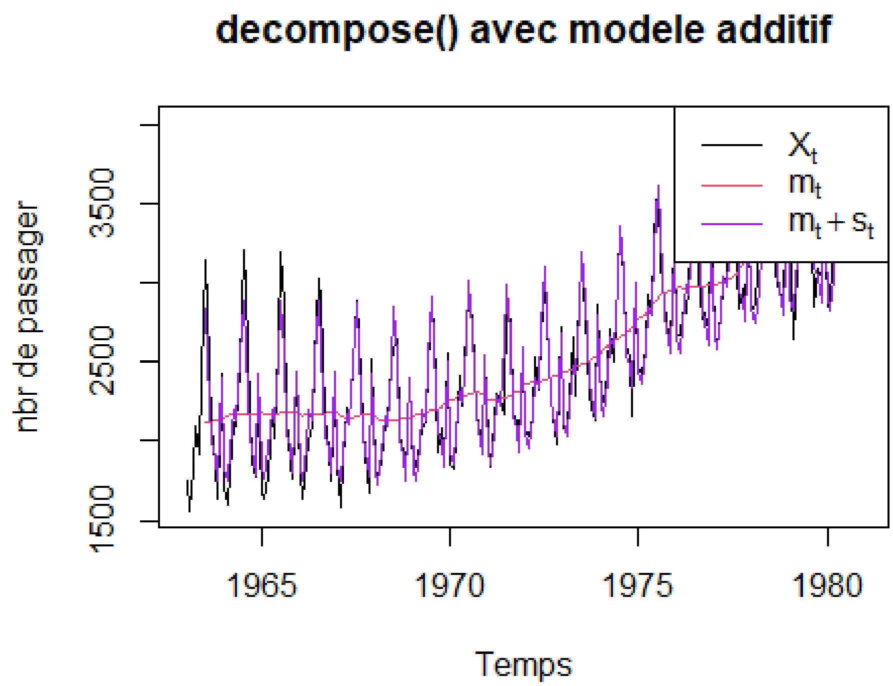
DECOMPOSITION DE LA SERIE TEMPORELLE :

```
fit1 <- decompose(X)
plot(fit1)
```



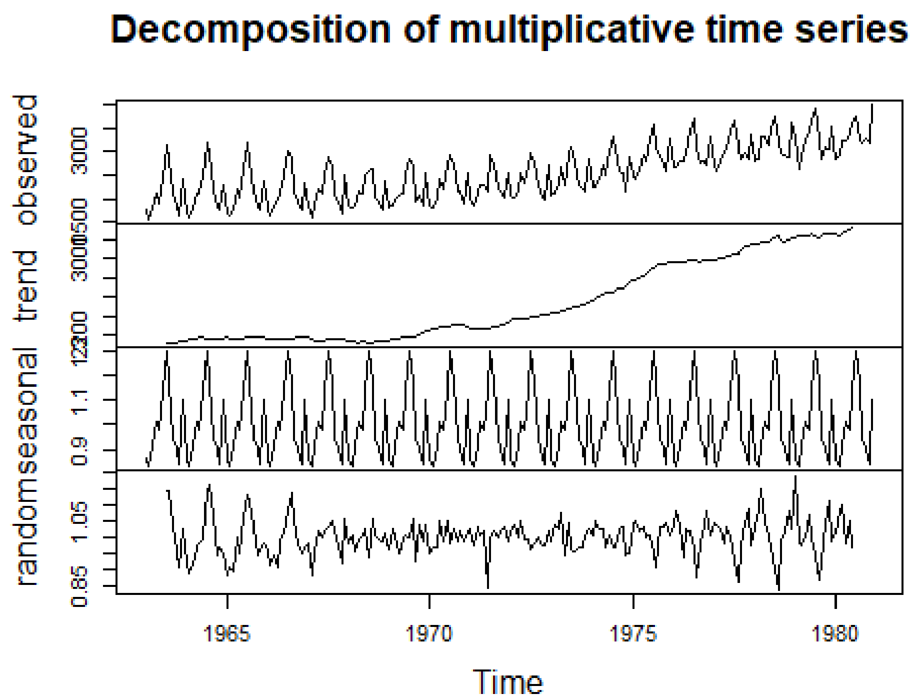
La fonction ‘decompose’ nous donne une stabilisation jusqu’à l’année 1970 puis une croissance bien remarquable, ainsi qu’un motif périodique.

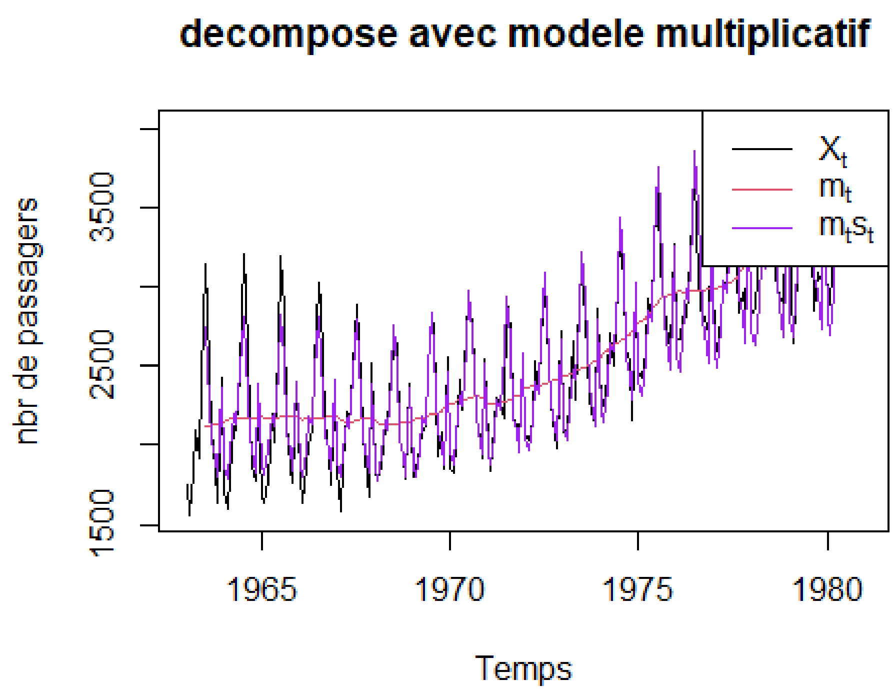
Cependant il semble que la variance des résidus ne dépend pas du temps ce qui suggère que le modèle proposé (modèle additif) est adéquat. Nous pouvons observer également la qualité de l’estimation en superposant les estimateurs de la tendance et de la saisonnalité au chronogramme.



On constate d'après le graphe que le modèle sous-estime l'amplitude des variations saisonnières jusqu'au 1968 alors que l'estimation après cette année est bonne.

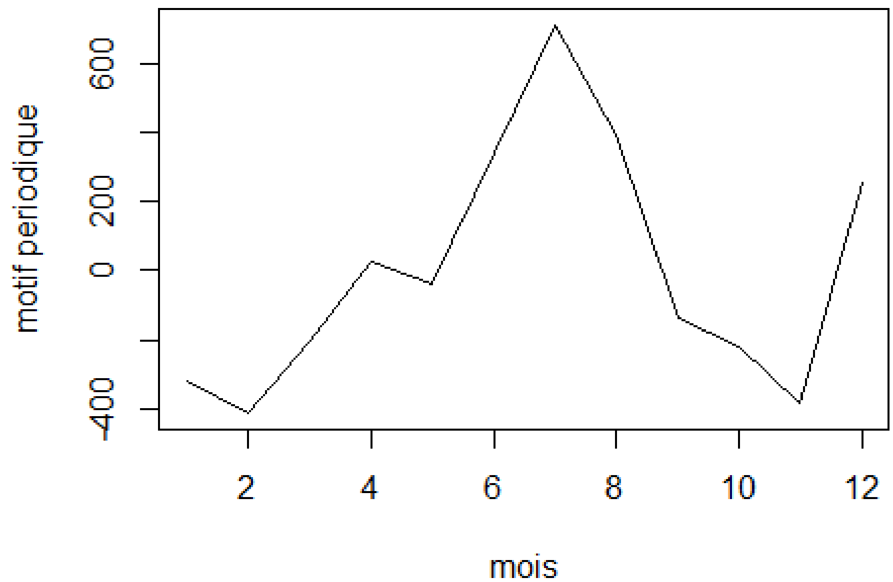
Jetons un œil sur le modèle multiplicatif :





Il n'y a pas de différence pour le décomposition, mais on voit bien que le modèle multiplicatif surestime la série après l'année 1972. L'estimation du modèle additif est mieux, donc on prend le modèle additif.

```
plot(fit1$figure,type="l",xlab="mois",ylab="motif periodique")
```



L'observation du motif de la composante saisonnière concorde avec une baisse importante de l'intérêt durant les mois 2 et 11 et très importante durant les mois d'été.

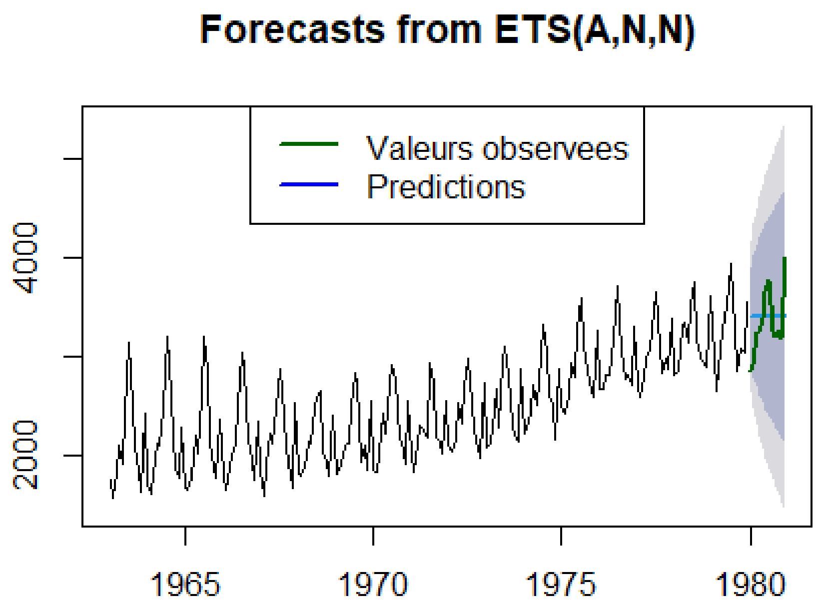
PREDICTION :

Pour évaluer la performance de prédiction, nous allons estimer les paramètres du modèle sur la série allant de janvier 1963 jusqu'à décembre 1979 et garder les observations de l'année 1980 pour les comparer avec les prévisions:

```
X.6379 <- window(X,start=1963,end=c(1979,12))
X.80 <- window(X,start=1980)
```

Commençons par un LISSAGE EXPONENTIEL SIMPLE:

```
fitLES = ets(X.6379,model="ANN")

predLES = forecast(fitLES,h=12)
plot(predLES)
points(X.80,type="l",col="darkgreen",lwd=2)
legend("top",c("Valeurs observees","Predictions"), col=c("darkgreen","blue"),
lty=rep(1,2),lwd = rep(2,2))
```



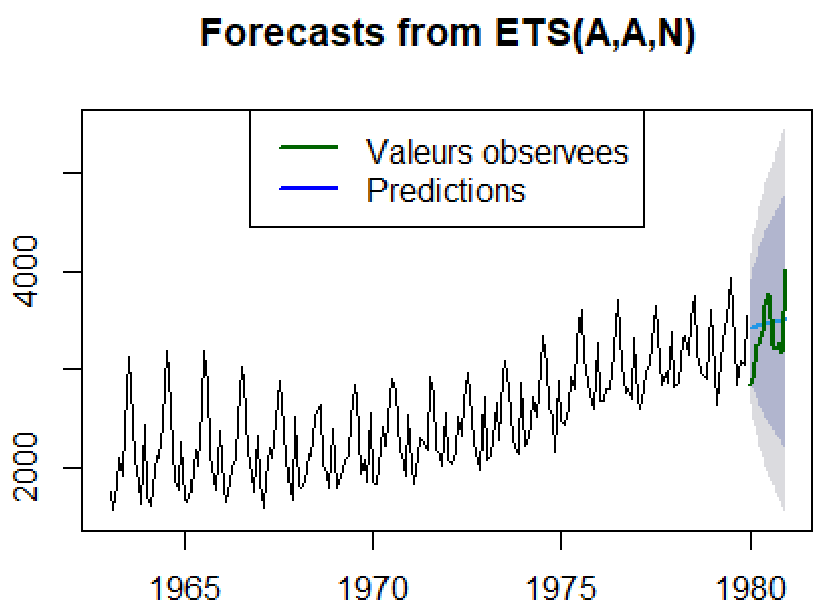

```
predict(fitLES,12)

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1980       3406.179 2912.789 3899.570 2651.604 4160.755
## Feb 1980       3406.179 2796.459 4015.899 2473.693 4338.666
## Mar 1980       3406.179 2699.014 4113.345 2324.663 4487.695
## Apr 1980       3406.179 2613.458 4198.901 2193.817 4618.542
## May 1980       3406.179 2536.276 4276.082 2075.777 4736.581
## Jun 1980       3406.179 2465.405 4346.953 1967.390 4844.969
## Jul 1980       3406.179 2399.512 4412.847 1866.614 4945.745
## Aug 1980       3406.179 2337.674 4474.685 1772.041 5040.318
## Sep 1980       3406.179 2279.224 4533.135 1682.650 5129.709
## Oct 1980       3406.179 2223.660 4588.699 1597.672 5214.687
## Nov 1980       3406.179 2170.592 4641.767 1516.511 5295.848
## Dec 1980       3406.179 2119.711 4692.648 1438.695 5373.663
```

La fonction predict donne les prédictions à l’horizon qu’on choisit. On observe que la prédiction est bien constante. La valeur 3406.179 n’est autre que la prévision à l’horizon 1 à partir de la dernière observation.

Passons au LISSAGE EXPONENTIEL DOUBLE :

```
fitLED = ets(X.6379,model="AAN")

predLED = forecast(fitLED,h=12)
plot(predLED)
points(X.80,type="l",col="darkgreen",lwd=2)
```

```r
legend("top",c("Valeurs observees","Predictions"), col=c("darkgreen","blue"),
lty=rep(1,2),lwd = rep(2,2))
```

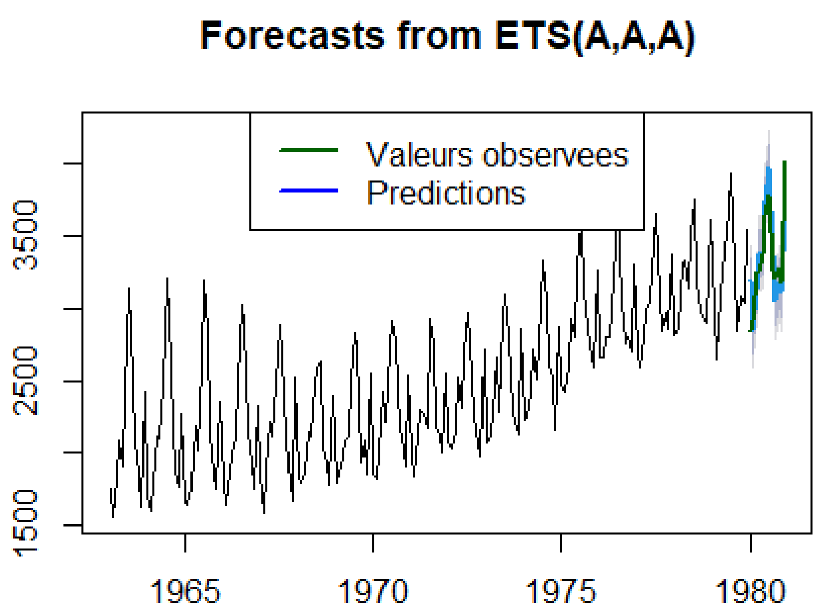


Ces deux lissages prédit mal nos donnée.

LISSAGE EXXPONENTIEL TRIPLE :

```r
fitHW <- ets(X.6379,model="AAA")

predHW <- forecast(fitHW,h=12)
plot(predHW)
points(X.80,type="l",col="darkgreen",lwd=2)
legend("top",c("Valeurs observees","Predictions"),col=c("darkgreen","blue"),
lty=rep(1,2),lwd = rep(2,2))

## ETS(A,A,A)
##
## Call:
##  ets(y = X.6379, model = "AAA")
##
##   Smoothing parameters:
##     alpha = 0.0928
##     beta  = 0.0046
##     gamma = 0.479
##
##   Initial states:
##     l = 2180.8865
```

```
##     b = 1.7331
##     s = 335.8738 -424.1515 -281.0965 -92.9194 504.7844 715.2654
##            315.7252 30.5647 20.1066 -256.943 -426.3811 -440.8287
##
##   sigma:  131.6432
##
##      AIC     AICc      BIC
## 3093.313 3096.603 3149.721
```

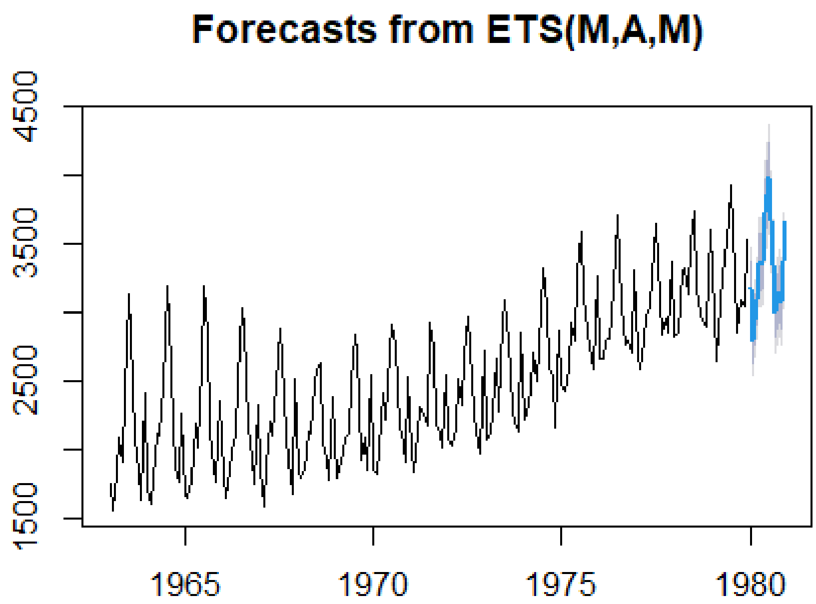


Ce modèle est plus pertinent visuellement, même s'il y a une surestimation mais il est mieux que les deux précédents.

COMPARAISON DES PREDICTION :

D'abord on fait une prédiction faite par R, ce modèle est sans tendance et avec une erreur et saisonnalité multiplicative :

```
fit <- ets(X.6379)
predfit <- forecast(fit,h=12)
plot(predfit)
```




```
summary(fit)

## ETS(M,A,M)
##
## Call:
##  ets(y = X.6379)
##
##   Smoothing parameters:
##     alpha = 0.0933
##     beta  = 0.0072
##     gamma = 0.5124
##
##   Initial states:
##     l = 2153.812
##     b = 8.9228
##     s = 1.0998 0.7854 0.8691 0.9817 1.3335 1.453
##            1.1438 0.9331 0.987 0.85 0.7461 0.8176
##
##   sigma:  0.0496
##
##      AIC     AICc      BIC
## 3059.861 3063.152 3116.269
##
## Training set error measures:
##                      ME      RMSE       MAE        MPE       MAPE       MASE
ACF1
```

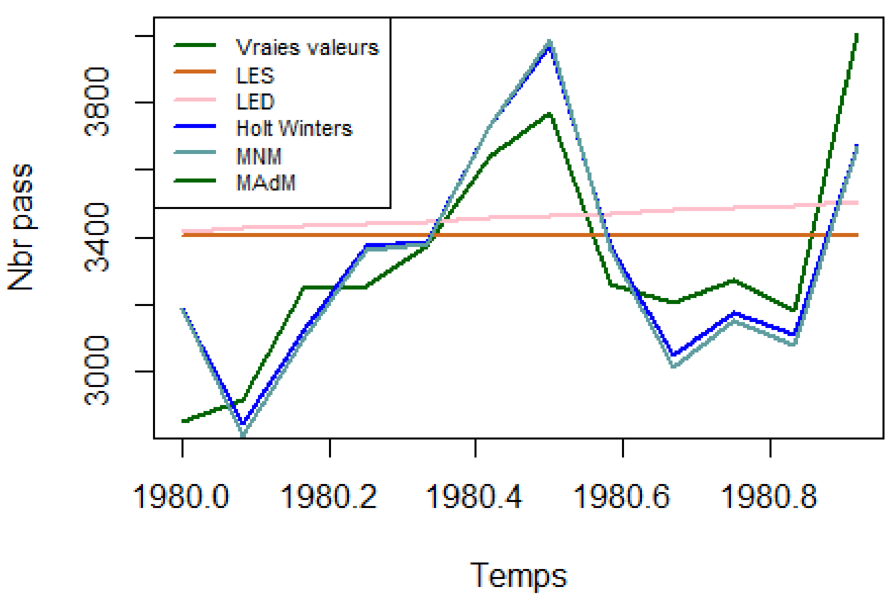
```
## Training set -7.621288 125.5826 89.80574 -0.269905 3.569428 0.7596573 0.21
87159
```

Je notice que l'AIC de ce modèle (=3059.861) est plus petite que celle du modèle du lissage exponentiel triple (=3093.313). Alors que c'est l'inverse pour AICc (=3063.152) et BIC (=3116.269) de ce modèle qui sont plus grand que ceux du modèle de lissage triple AICc (=3096.603) et BIC (=3149.721).

Il est claire qu'il y a une ressemblance entre cette prédiction et celle du lissage exponentiel triple.

Maintenant comparent toutes les prédictions faites:

```
plot(X.80,col="darkgreen",lwd=2,ylab="Nbr pass",xlab="Temps")
points(predLES$mean,col="chocolate",lwd=2,type="l")
points(predLED$mean,col="pink",lwd=2,type="l")
points(predHW$mean,col="blue",lwd=2,type="l")
points(predfit$mean,col="cadetblue",lwd=2,type="l")
legend("topleft",c("Vraies valeurs","LES","LED","Holt Winters","MNM","MAdM"),
         col=c("darkgreen","chocolate","pink","blue","cadetblue"),lty=rep(1,6
),lwd=rep(2,6),cex=0.7)
```



D'après le graphe, il est claire que la prédiction du model de lissage exponentiel triple qui est le plus proche et plus convenable. Si on se base sur l'AIC le plus petit pour choisir

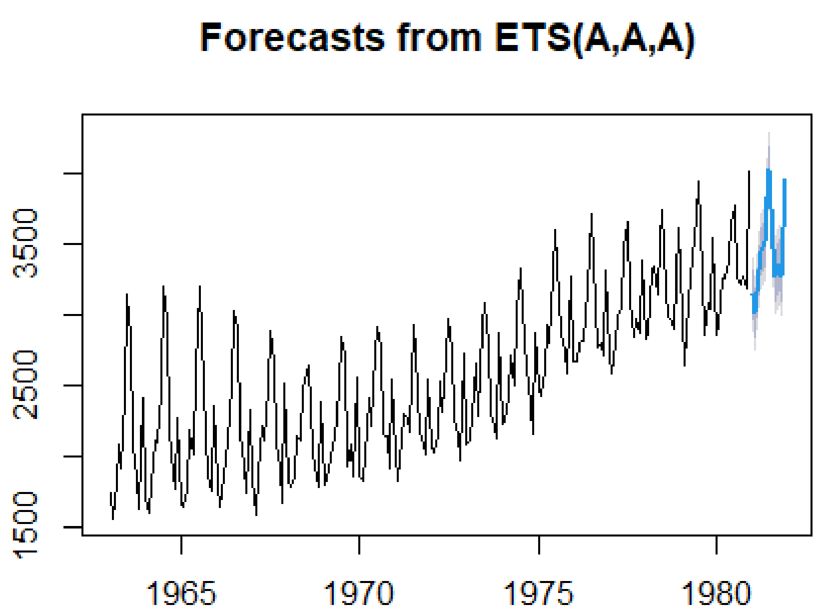
```
fit$aicc

## [1] 3063.152
```

```
fitHW$aicc

## [1] 3096.603

fitLES$aicc

## [1] 3517.924

fitLED$aicc

## [1] 3522.316
```

Donc le modèle donné par R est le plus convenable, en fait il n'y a pas grande différence entre les deux concernant la prédiction. Je choisis le modèle de lissage exponentiel triple.

PREDICTION DE L'ANNEE APRES (1981) :

```
fittotal <- ets(X,model="AAA")

predfittotal <- forecast(fittotal,h=12)
plot(predfittotal)
```

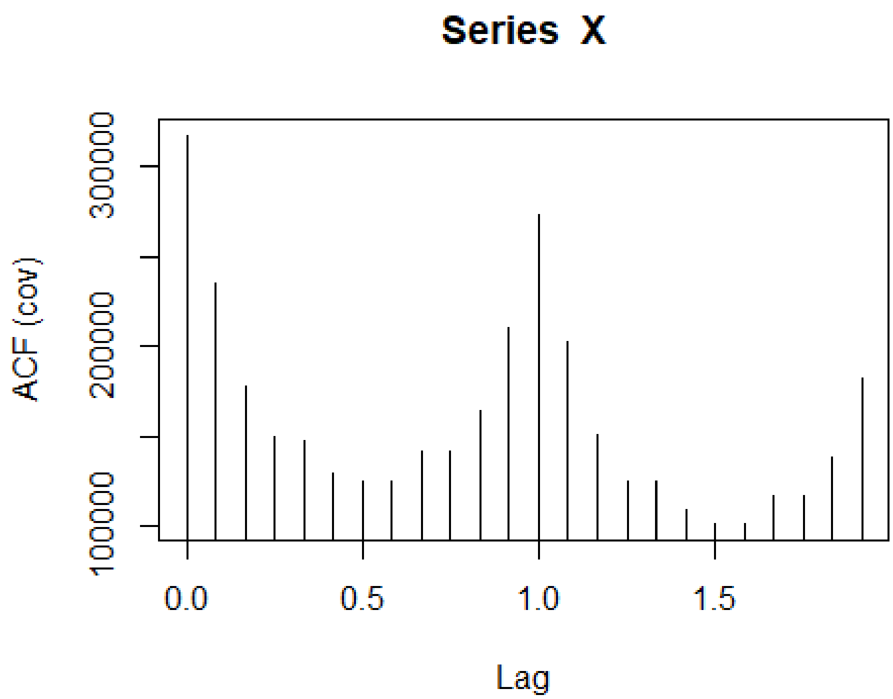


MODELISATION

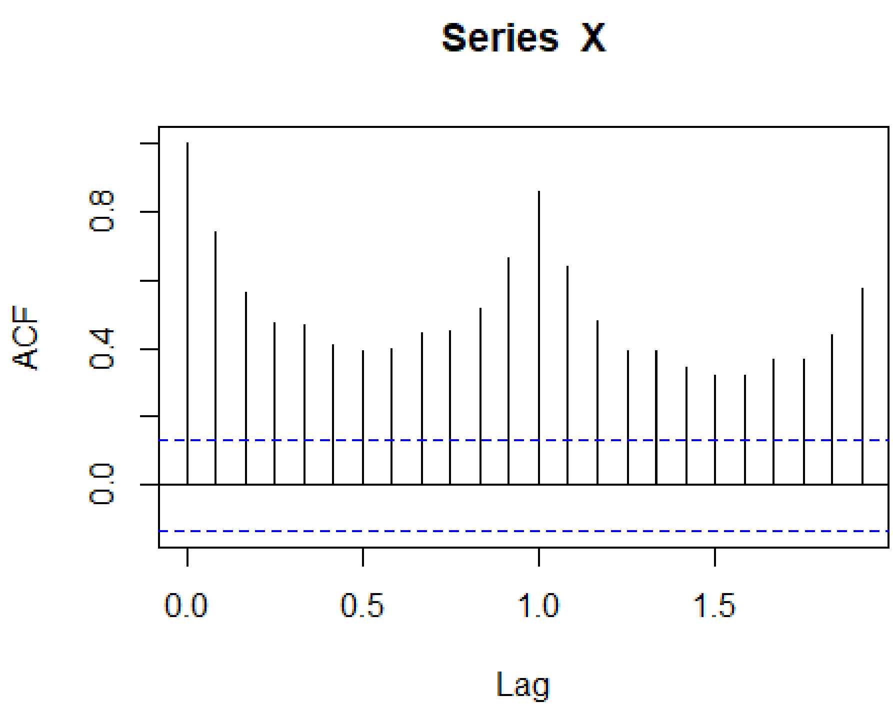
Estimation de la moyenne et des fonctions d'autocovariance et d'autocorrélation:

```
mean(X)

## [1] 2547.13
```

```
acf(X,type ="covariance")
```




```
acf(X,type ="correlation")
```



La saisonnalité est toujours bien claire, les valeur sont toutes positives.

Elle atteint une valeur maximal locale, ce qui signifie que certaines périodes de l'année sont fortement corrélées au même périodes des années précédents.

Typiquement les gens voyagent chaque année souvent l'été plus que les autres saisons.

On notice que la série n'est pas stationnaire, car son corrélogramme ne décroît pas rapidement vers 0.

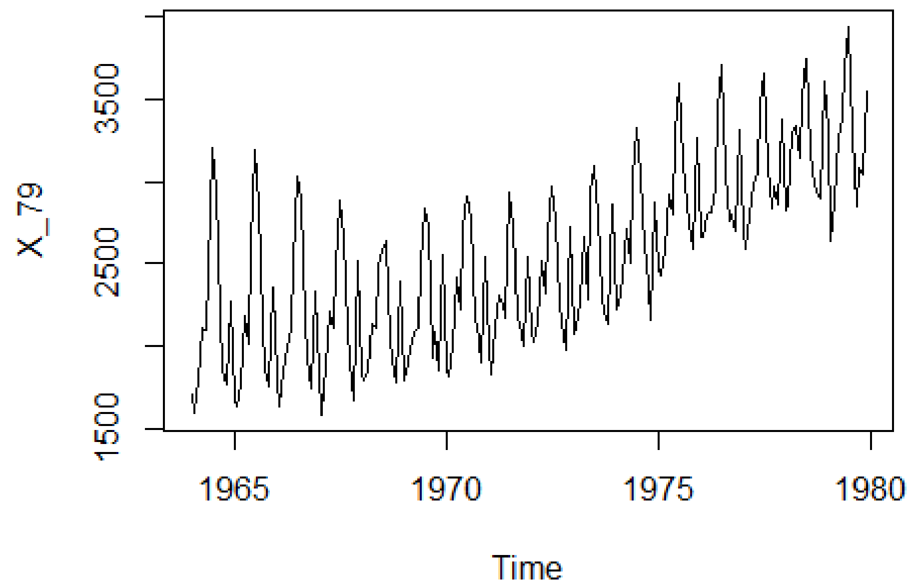
Faisant un test statistique 'test de blancheur' pour valider que notre série n'est pas stationnaire:

```
length(X)

## [1] 216

Box.test(X,lag=20,type="Box-Pierce")

##
##  Box-Pierce test
##
## data:  X
## X-squared = 1089.9, df = 20, p-value < 2.2e-16
```

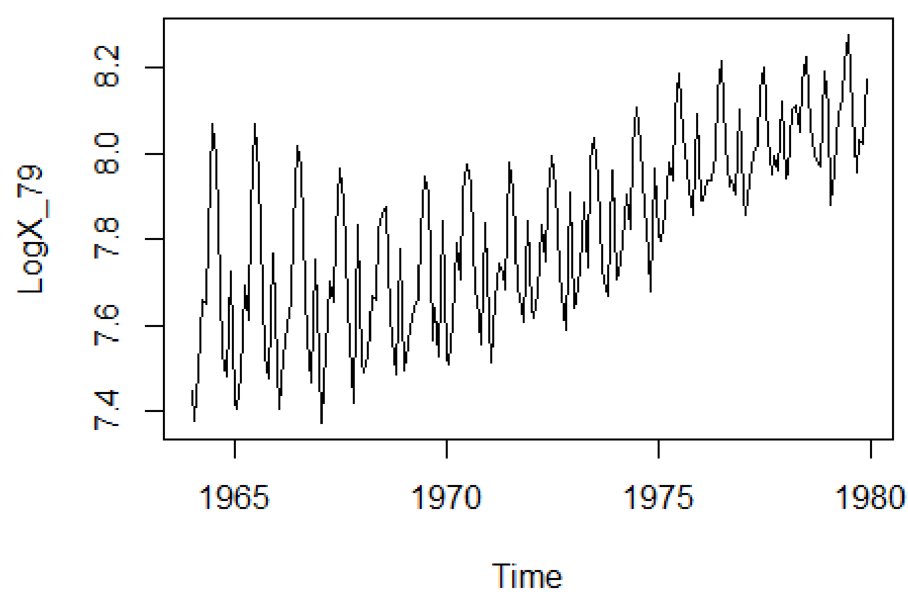
La p-valeur est inférieur à 5%, ce qui signifie que le bruit n'est pas blanc, donc la série n'est pas stationnaire.

On doit modifier ou transformer notre série :

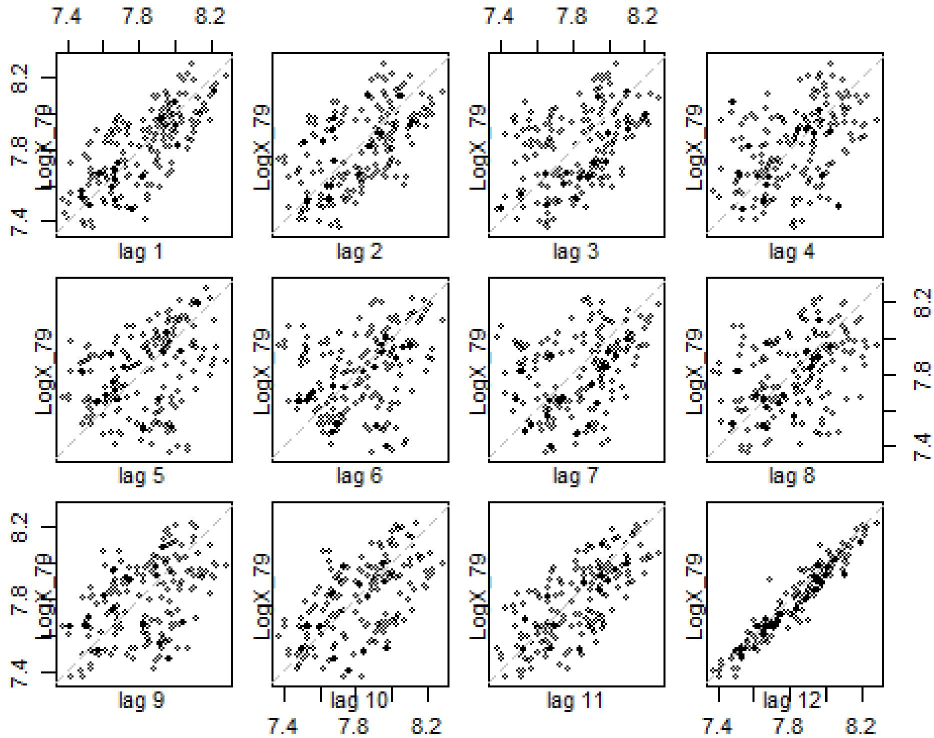
```
X_79=window(X,start=1964,end=c(1979,12))
plot(X_79)
```



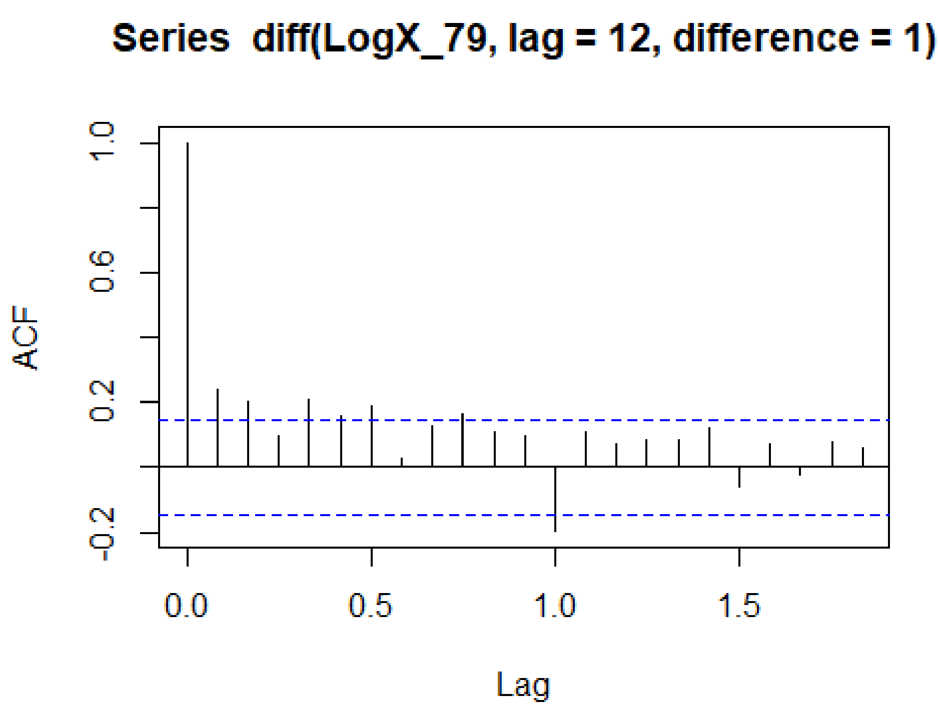

```
LogX_79=log(X_79)
plot(LogX_79)
```



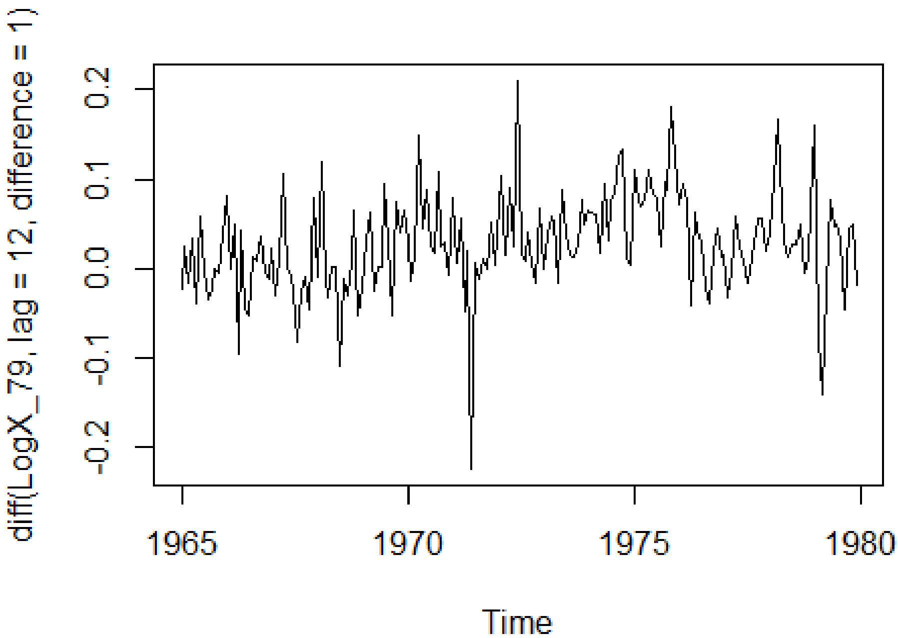
```r
lag.plot(LogX_79,lags=12,layout=c(3,4),do.lines=FALSE)
```




```r
acf(diff(LogX_79,lag=12,difference=1))
```



```
plot(diff(LogX_79,lag=12,difference=1))
```

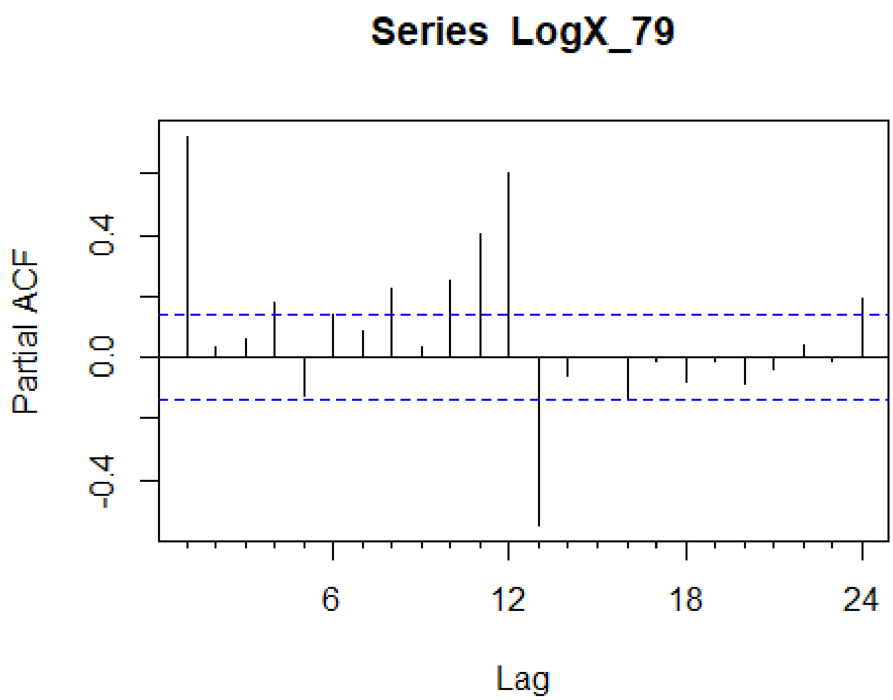


Je notice que la saisonnalité a disparu ça ressemble de plus en plus à un processus stationnaire, la tendance est éliminé comme le montre le corrélogramme.

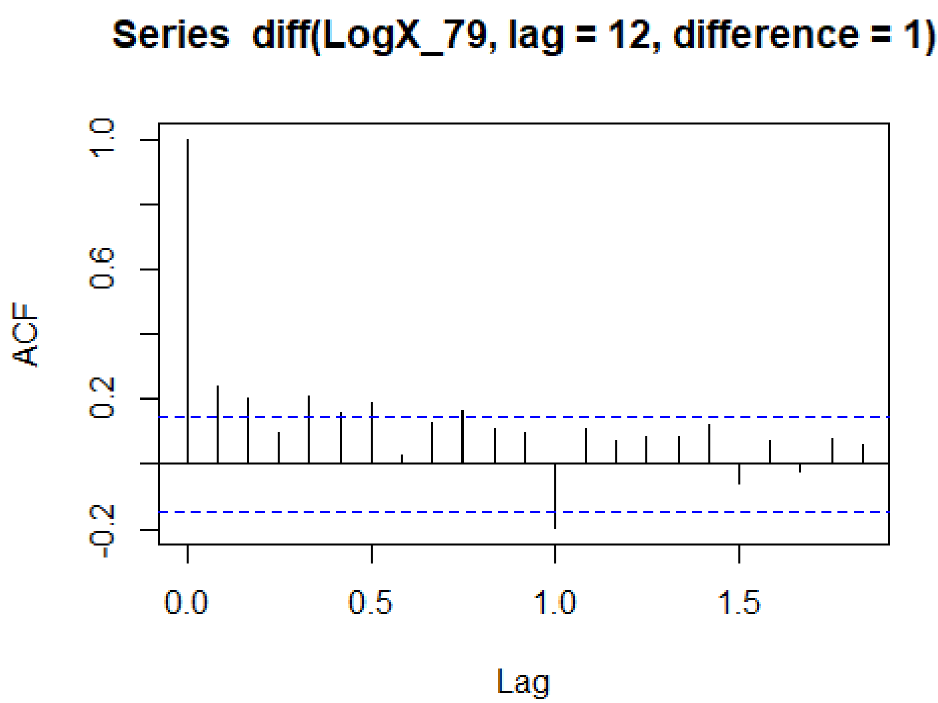
Maintenant on passe à la modélisation des informations restées après avoir enlevé la saisonnalité et la tendance.

Affichons le ACF et PACF pour savoir la nature de notre modèle

```
Pacf(LogX_79)
```




```
acf(diff(LogX_79,lag=12,difference=1))
```



D’après le ACF, on remarque une décroissance exponentielle, donc ce n’est pas un MA.

Pour le PACF, on notice une variance et il n'y a pas de décroissance donc ce n'est pas un AR Donc notre modèle est un ARMA.

```
model1=Arima(LogX_79,order=c(2,0,2))
model1

## Series: LogX_79
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1      ma2    mean
##       1.3411  -0.3436  -0.7613  -0.1585  7.8443
## s.e.  0.1229   0.1225   0.1185   0.1055  0.1662
##
## sigma^2 = 0.01917:  log likelihood = 108.71
## AIC=-205.42   AICc=-204.97   BIC=-185.88

t_stat(model1)

##             ar1       ar2       ma1       ma2 intercept
## t.stat 10.91279 -2.805570 -6.426749 -1.502134  47.18601
## p.val   0.00000  0.005023  0.000000  0.133063   0.00000
```

D'après la fonction t_stat, on constate qu'il y a un problème avec une variable de MA, donc on va modifier notre modèle :

```
model2=Arima(LogX_79,order=c(2,0,1))
model2

## Series: LogX_79
## ARIMA(2,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1    mean
##       1.4748  -0.4769  -0.9356  7.8411
## s.e.  0.0694   0.0691   0.0207  0.1678
##
## sigma^2 = 0.01928:  log likelihood = 107.71
## AIC=-205.42   AICc=-205.09   BIC=-189.13

t_stat(model2)

##             ar1       ar2       ma1 intercept
## t.stat 21.25154 -6.898009 -45.30214  46.73578
## p.val   0.00000  0.000000   0.00000   0.00000
```

Maintenant vérifiant la corrélation:

```
cor.arma(model2)

##                   ar1          ar2         ma1    intercept
## ar1       1.000000000 -0.999200406 -0.37767914 -0.004460055
```

```
## ar2       -0.999200406  1.000000000  0.36182394  0.007934723
## ma1       -0.377679136  0.361823945  1.00000000 -0.019348926
## intercept -0.004460055  0.007934723 -0.01934893  1.000000000
```

il y a une forte corrélation avec une variable de AR, modifions une autre fois encore le modèle:

```
model3=Arima(LogX_79,order=c(1,0,1))
model3

## Series: LogX_79
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ma1    mean
##       0.7731  -0.0705  7.8108
## s.e.  0.0689   0.1150  0.0421
##
## sigma^2 = 0.02134:  log likelihood = 98.02
## AIC=-188.04   AICc=-187.82   BIC=-175.01

 t_stat(model3)

##             ar1       ma1 intercept
## t.stat 11.22549 -0.613407  185.3952
## p.val   0.00000  0.539608    0.0000

  cor.arma(model3)

##                     ar1         ma1    intercept
## ar1        1.000000000 -0.73544581 -0.009688607
## ma1       -0.735445810  1.00000000  0.011123907
## intercept -0.009688607  0.01112391  1.000000000
```

Le problème de corrélation est réglé mais d'après t_stat il y a un problème avec la variable de MA

```
model4=Arima(LogX_79,order=c(1,0,0))
model4

## Series: LogX_79
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1    mean
##       0.7405  7.8110
## s.e.  0.0493  0.0398
##
## sigma^2 = 0.02127:  log likelihood = 97.82
## AIC=-189.65   AICc=-189.52   BIC=-179.88

 t_stat(model4)
```

```
##             ar1 intercept
## t.stat 15.00903  196.4399
## p.val   0.00000    0.0000
```

```
  cor.arma(model4)
```

```
##                   ar1    intercept
## ar1        1.000000000 -0.001556844
## intercept -0.001556844  1.000000000
```

Passons au vérification du bruit blanc

```
Box.test(model1$residuals, lag = 20, type = "Box-Pierce")
```

```
## 
##  Box-Pierce test
## 
## data:  model1$residuals
## X-squared = 200.82, df = 20, p-value < 2.2e-16
```

```
Box.test(model2$residuals, lag = 20, type = "Box-Pierce")
```

```
## 
##  Box-Pierce test
## 
## data:  model2$residuals
## X-squared = 197.4, df = 20, p-value < 2.2e-16
```

```
Box.test(model3$residuals, lag = 20, type = "Box-Pierce")
```

```
## 
##  Box-Pierce test
## 
## data:  model3$residuals
## X-squared = 175.58, df = 20, p-value < 2.2e-16
```

```
Box.test(model4$residuals, lag = 20, type = "Box-Pierce")
```

```
## 
##  Box-Pierce test
## 
## data:  model4$residuals
## X-squared = 177.65, df = 20, p-value < 2.2e-16
```

Je notice que tous les modèles précédents ne sont pas bons.

Maintenant j'utilise auto.arima pour voir les possibilités que R nous propose:

```
auto.arima(LogX_79)
```

```
## Series: LogX_79
## ARIMA(1,1,3)(0,1,1)[12]
##
```

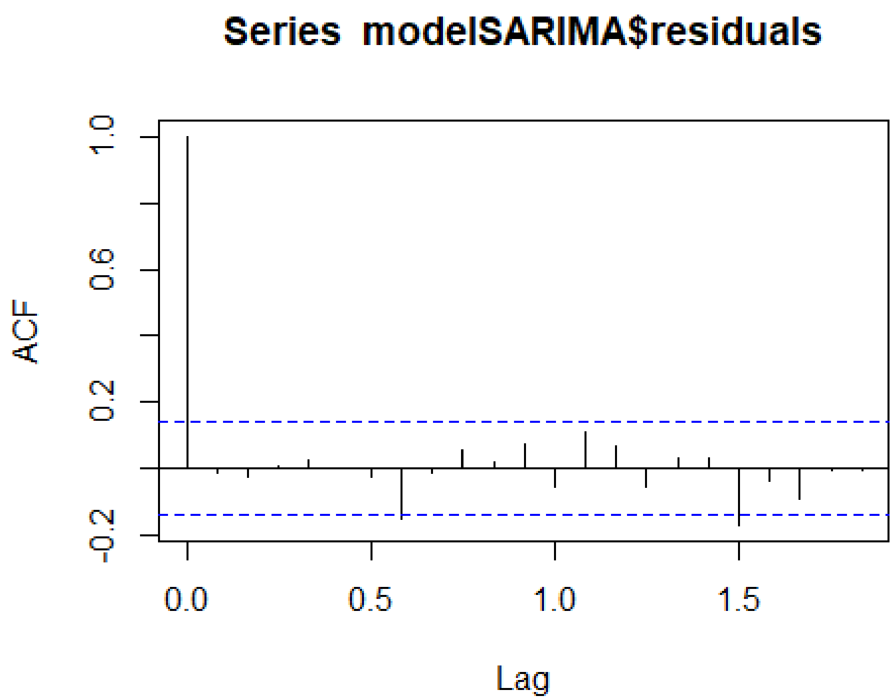
```
## Coefficients:
##           ar1     ma1      ma2      ma3     sma1
##       -0.8072  0.0955  -0.6467  -0.2143  -0.4599
## s.e.   0.1265  0.1438   0.1103   0.0769   0.0687
##
## sigma^2 = 0.002424:  log likelihood = 285.49
## AIC=-558.98   AICc=-558.49   BIC=-539.85
```

Donc comme le modèle ARMA n'est pas compatible avec notre série, essayons ARMA saisonnier SARIMA:

```
modelSARIMA=auto.arima(LogX_79)
modelSARIMA

## Series: LogX_79
## ARIMA(1,1,3)(0,1,1)[12]
##
## Coefficients:
##           ar1     ma1      ma2      ma3     sma1
##       -0.8072  0.0955  -0.6467  -0.2143  -0.4599
## s.e.   0.1265  0.1438   0.1103   0.0769   0.0687
##
## sigma^2 = 0.002424:  log likelihood = 285.49
## AIC=-558.98   AICc=-558.49   BIC=-539.85

t_stat(modelSARIMA)

##              ar1      ma1       ma2       ma3      sma1
## t.stat -6.381491 0.664564 -5.861061 -2.786883 -6.694794
## p.val   0.000000 0.506329  0.000000  0.005322  0.000000

cor.arma(modelSARIMA)

##              ar1        ma1          ma2         ma3          sma1
## ar1   1.00000000 -0.8379956  0.874043718  0.05818682  0.063807319
## ma1  -0.83799556  1.0000000 -0.732662507 -0.43635552 -0.194098743
## ma2   0.87404372 -0.7326625  1.000000000  0.01538722  0.009050263
## ma3   0.05818682 -0.4363555  0.015387220  1.00000000  0.188832168
## sma1  0.06380732 -0.1940987  0.009050263  0.18883217  1.000000000

Box.test(modelSARIMA$residuals,lag=20)

##
##  Box-Pierce test
##
## data:  modelSARIMA$residuals
## X-squared = 18.756, df = 20, p-value = 0.5377
```

Tout est bien, la corrélation et le test du bruit blanc, la P_valeur est supérieur à 5%, ce qui valide le model.

```
acf(modelSARIMA$residuals)
```



Notre modèle est mieux que les précédents. Voyons si on peut avoir un meilleur model :

```
modelSARIMA1 = Arima(LogX_79, order =c(1,1,1), seasonal = list(order=c(0,1,1)
, period =12))
modelSARIMA1
```

```
## Series: LogX_79
## ARIMA(1,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.1495  -0.8896  -0.4427
## s.e.  0.0872   0.0407   0.0681
##
## sigma^2 = 0.002462:  log likelihood = 283.2
## AIC=-558.41   AICc=-558.18   BIC=-545.66
```

```
t_stat(modelSARIMA1)
```

```
##             ar1       ma1      sma1
## t.stat 1.713701 -21.87934 -6.498093
## p.val  0.086584   0.00000  0.000000
```

```
cor.arma(modelSARIMA1)
```

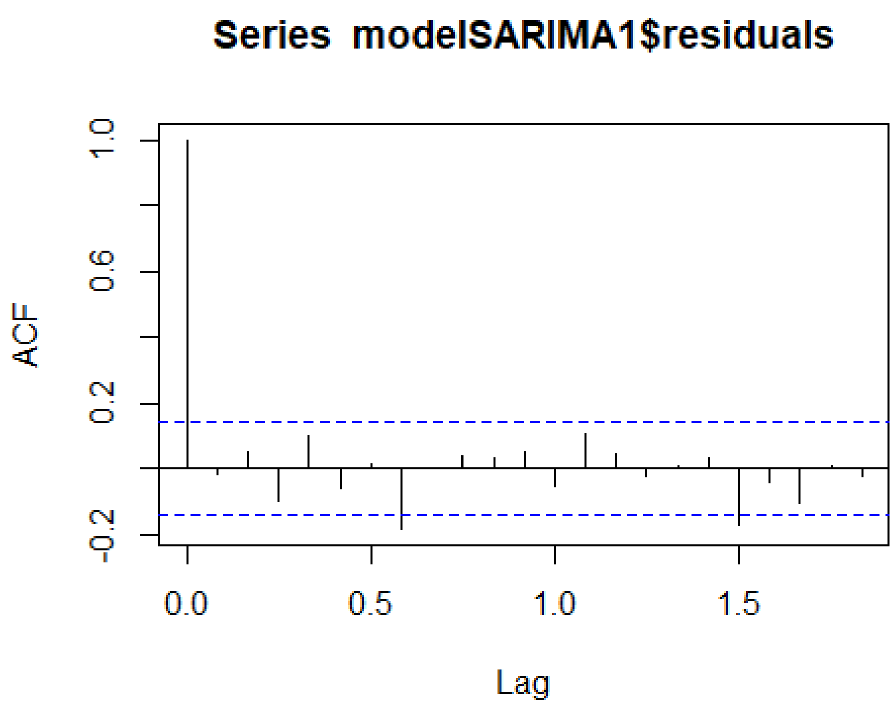
```
##             ar1         ma1        sma1
## ar1   1.0000000 -0.48327315 -0.18684490
```

```
## ma1  -0.4832732  1.00000000 -0.05317942
## sma1 -0.1868449 -0.05317942  1.00000000
```

```
Box.test(modelSARIMA1$residuals,lag=20)
```

```
##
##  Box-Pierce test
##
## data:  modelSARIMA1$residuals
## X-squared = 24.058, df = 20, p-value = 0.2399
```

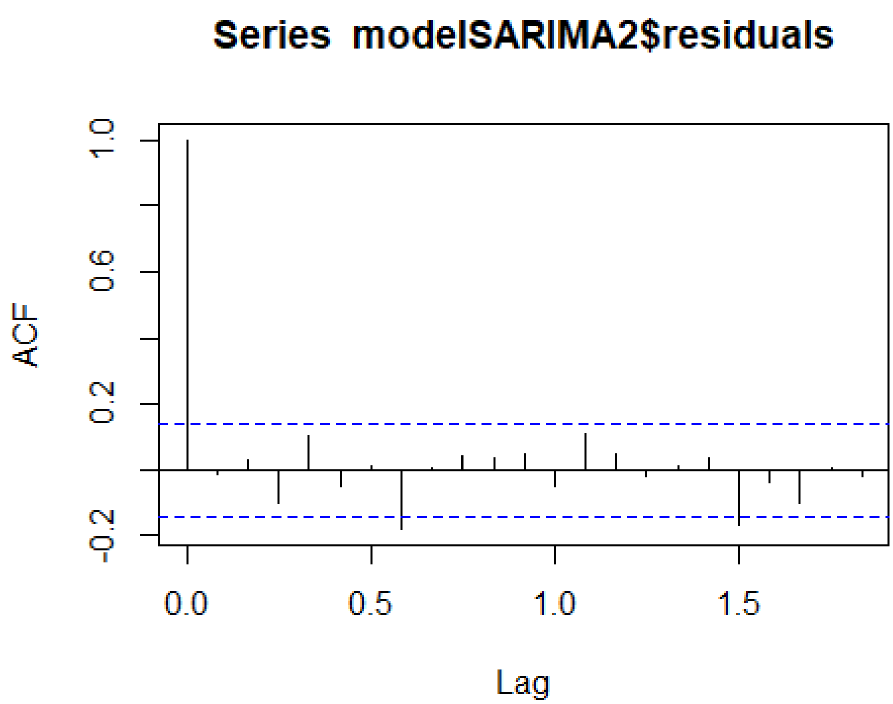
```
acf(modelSARIMA1$residuals)
```




```
modelSARIMA2 = Arima(LogX_79, order =c(1,1,2), seasonal = list(order=c(0,1,1)
, period =12))
modelSARIMA2
```

```
## Series: LogX_79
## ARIMA(1,1,2)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     ma2     sma1
##       0.3602  -1.1023  0.1829  -0.4435
## s.e.  0.4116   0.4256  0.3685   0.0683
##
## sigma^2 = 0.002472:  log likelihood = 283.33
## AIC=-556.66   AICc=-556.32   BIC=-540.73
```

```
t_stat(modelSARIMA2)

##              ar1      ma1      ma2     sma1
## t.stat 0.875072 -2.59026 0.496402 -6.49084
## p.val  0.381535  0.00959 0.619611  0.00000

cor.arma(modelSARIMA2)

##                ar1        ma1         ma2         sma1
## ar1   1.000000000 -0.9842329  0.98009291  0.007234611
## ma1  -0.984232947  1.0000000 -0.99743626 -0.049591800
## ma2   0.980092913 -0.9974363  1.00000000  0.046145721
## sma1  0.007234611 -0.0495918  0.04614572  1.000000000

Box.test(modelSARIMA2$residuals,lag=20)

##
##  Box-Pierce test
##
## data:  modelSARIMA2$residuals
## X-squared = 23.037, df = 20, p-value = 0.287

acf(modelSARIMA2$residuals)
```



La corrélation dans ce dernier modèle est très forte pour ma1 et ar1.

Je vais choisir entre le modèle modelSARIMA et modelSARIMA1:

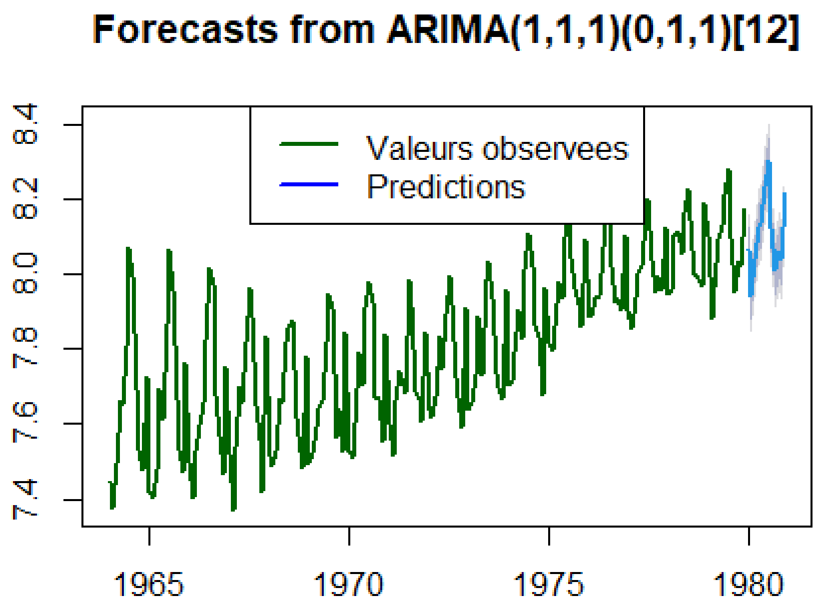
```
modelSARIMA$aic
```

```
## [1] -558.9764

modelSARIMA1$aic

## [1] -558.4094

modelSARIMA$aicc

## [1] -558.488

modelSARIMA1$aicc

## [1] -558.1795

modelSARIMA$bic

## [1] -539.8521

modelSARIMA1$bic

## [1] -545.6598
```

En se basant sur le AIC et AICc le model le plus pertinent est le model modelSARIMA1.

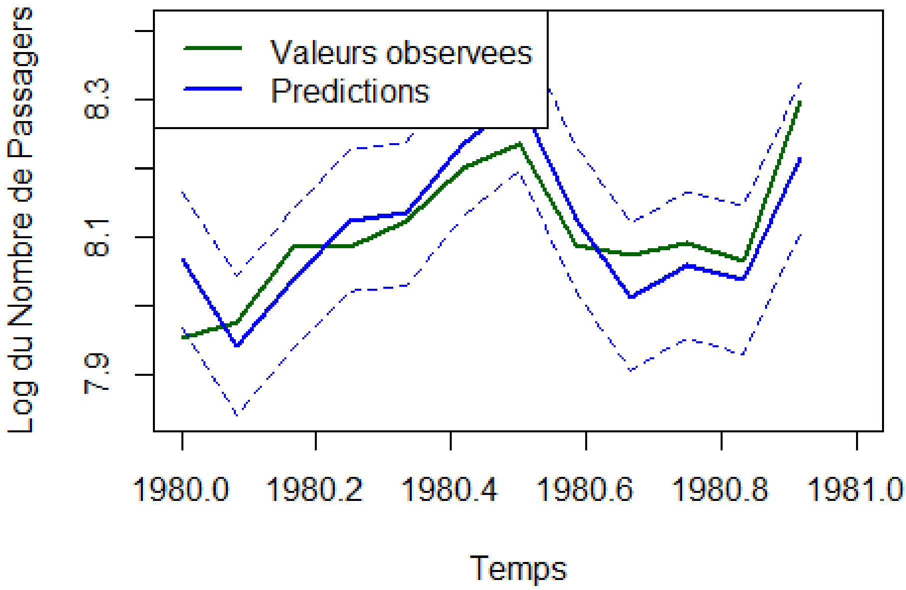
PREDICTION

```
predSARIMA=forecast(modelSARIMA1,12)
predSARIMA

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1980       8.066547 8.002955 8.130138 7.969292 8.163801
## Feb 1980       7.942566 7.876862 8.008270 7.842080 8.043051
## Mar 1980       8.038919 7.972533 8.105305 7.937390 8.140448
## Apr 1980       8.124465 8.057544 8.191385 8.022118 8.226811
## May 1980       8.133609 8.066177 8.201040 8.030481 8.236736
## Jun 1980       8.234323 8.166387 8.302258 8.130424 8.338221
## Jul 1980       8.300431 8.231995 8.368866 8.195767 8.405094
## Aug 1980       8.127123 8.058191 8.196055 8.021700 8.232545
## Sep 1980       8.013302 7.943878 8.082727 7.907126 8.119478
## Oct 1980       8.058770 7.988856 8.128684 7.951846 8.165694
## Nov 1980       8.037810 7.967410 8.108209 7.930143 8.145477
## Dec 1980       8.214274 8.143392 8.285156 8.105869 8.322679

plot(predSARIMA)
points(LogX_79,type="l",col="darkgreen",lwd=2)
legend("top",c("Valeurs observees","Predictions"),col=c("darkgreen","blue"),
 lty=rep(1,2),lwd = rep(2,2))
```



PREVISION:

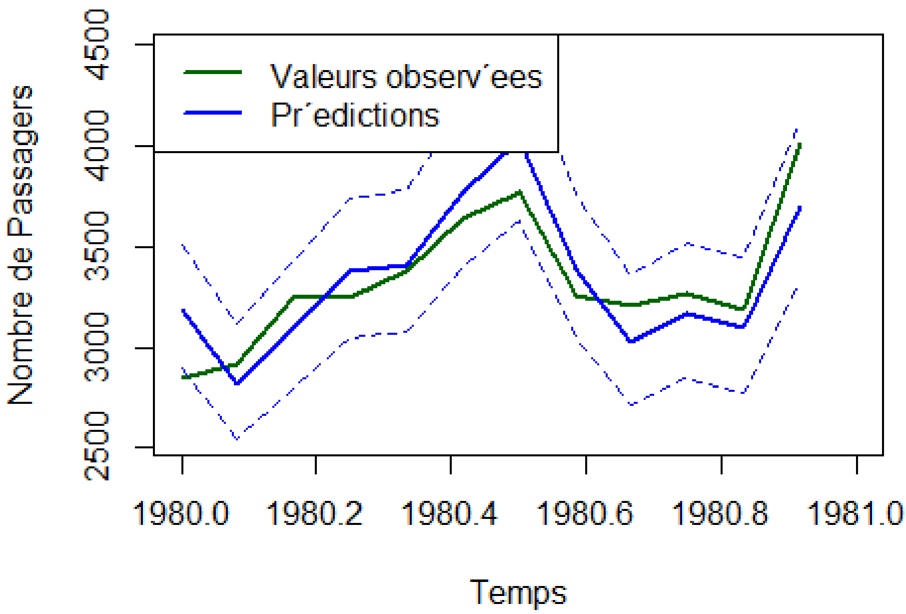
```
LogX_80 = log(X.80)
plot(LogX_80,col="darkgreen",lwd=2,ylab="Log du Nombre de Passagers",xlab="Te
mps",xlim=c(1980,1981),
 ylim=range(c(LogX_80,predSARIMA$lower,predSARIMA$upper)))
points(predSARIMA$mean,col="blue",lwd=2,type="l")
points(predSARIMA$lower[,2],col="blue",type="l",lty=2)
points(predSARIMA$upper[,2],col="blue",type="l",lty=2)
legend("topleft",c("Valeurs observees","Predictions"),
col=c("darkgreen","blue"),lty=rep(1,2),lwd = rep(2,2))
```



On remarque que la prédiction est très proche des valeurs observées. Ce modèle est pertinent

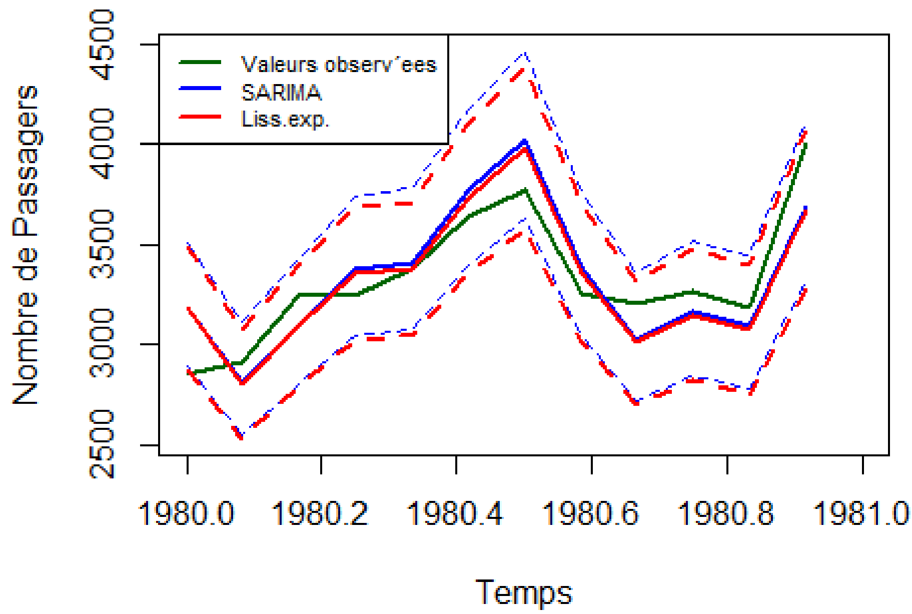
Finalement, retournant au données d'origine par passage à l'exponentielle.

```
plot(X.80,col="darkgreen",lwd=2,ylab="Nombre de Passagers",xlab="Temps",
xlim=c(1980,1981),ylim=range(c(X.80,exp(predSARIMA$lower),
 exp(predSARIMA$upper))))
points(exp(predSARIMA$mean),col="blue",lwd=2,type="l")
points(exp(predSARIMA$lower[,2]),col="blue",type="l",lty=2)
points(exp(predSARIMA$upper[,2]),col="blue",type="l",lty=2)
legend("topleft",c("Valeurs observ´ees","Pr´edictions"),
 col=c("darkgreen","blue"),lty=rep(1,2),lwd = rep(2,2))
```



Faisant une dernière comparaison entre SARIMA et lissage exponentielle:

```
fit=ets(X.6379)
predEts=forecast(fit,12)
plot(X.80,col="darkgreen",lwd=2,ylab="Nombre de Passagers",xlab="Temps",
xlim=c(1980,1981),ylim=range(c(X.80,
exp(predSARIMA$lower),exp(predSARIMA$upper),predEts$lower,predEts$upper)))
points(exp(predSARIMA$mean),col="blue",lwd=2,type="l")
points(exp(predSARIMA$lower[,2]),col="blue",type="l",lty=2)
points(exp(predSARIMA$upper[,2]),col="blue",type="l",lty=2)
points(predEts$mean,col="red",lwd=2,type="l")
points(predEts$lower[,2],col="red",lwd=2,type="l",lty=2)
points(predEts$upper[,2],col="red",lwd=2,type="l",lty=2)
legend("topleft",c("Valeurs observ´ees","SARIMA", "Liss.exp."),
 col=c("darkgreen","blue","red"),lty=rep(1,3),lwd = rep(2,3),cex=0.7)
```



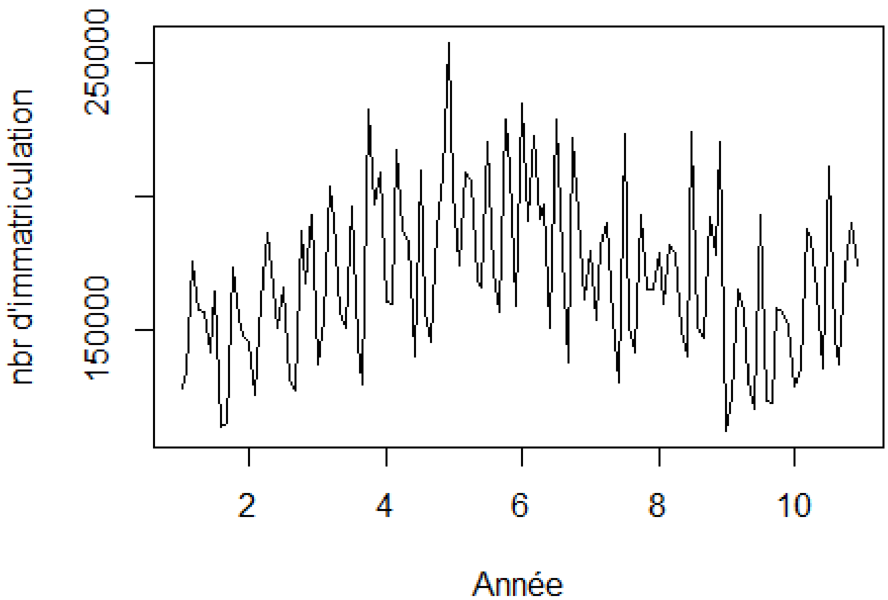
Conclusion:

SARIMA et Lissage Exponentiel sont très semblables, mais je prendrais le Lissage Exponentiel qui s'approche un peu plus des valeurs réelles.

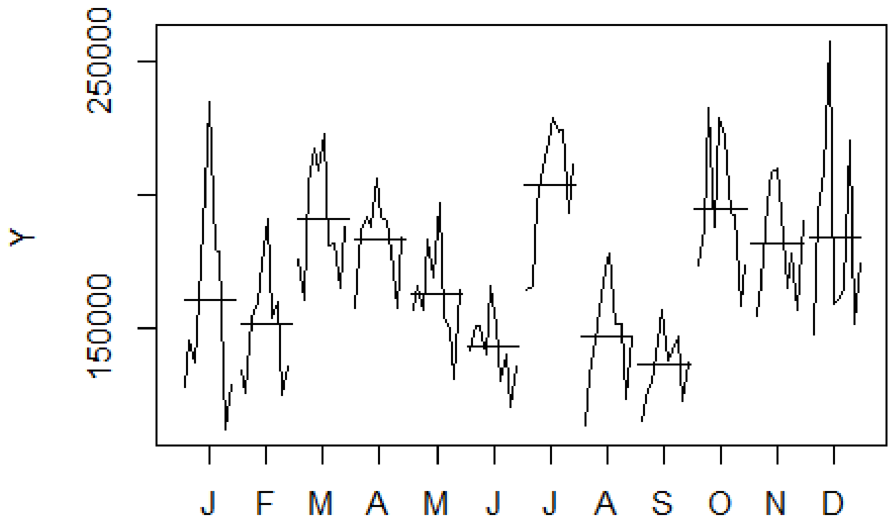
Nombre d'immatriculation en France:

Je m'intéresse dans ce jeu de données à l'évolution sur 10 ans du nombre d'immatriculations de voitures particulières en France

Traçons la courbe de la série :



Je notice une variance qui n'est pas constante. Il y a une saisonnalité qui n'est pas trop claire. Il n'y a pas de linéarité.

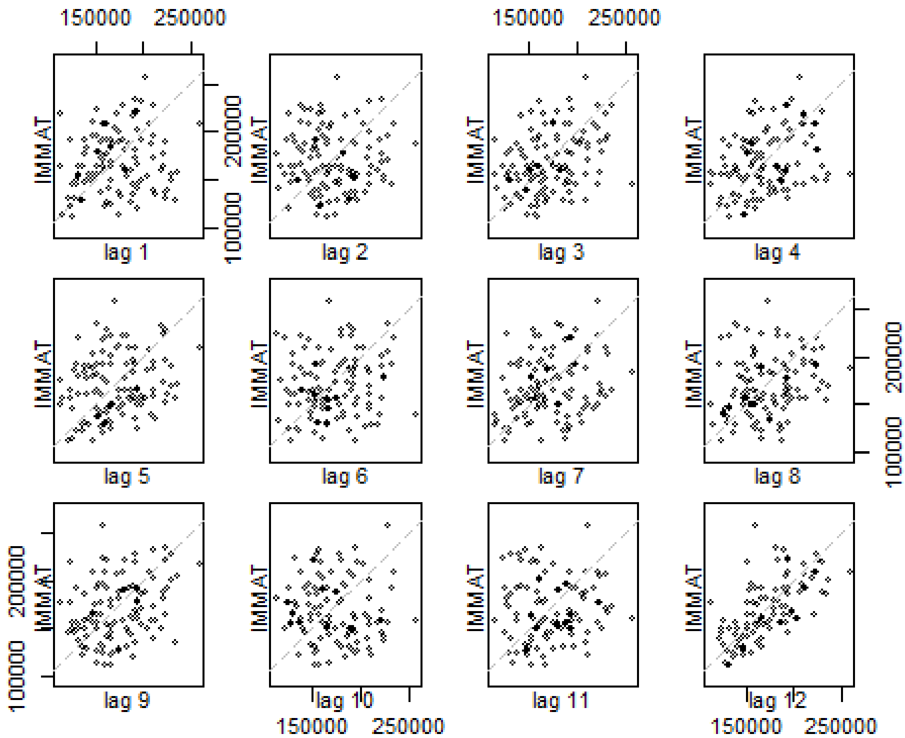


Le monthplot :

On remarque que le nombre des immatriculations atteint son maximum en mois de décembre, il est aussi important en janvier juillet et octobre.

Alors qu'il est très petit en juin, aout et septembre. Aussi il y a une saisonnalité.

Le lag out :

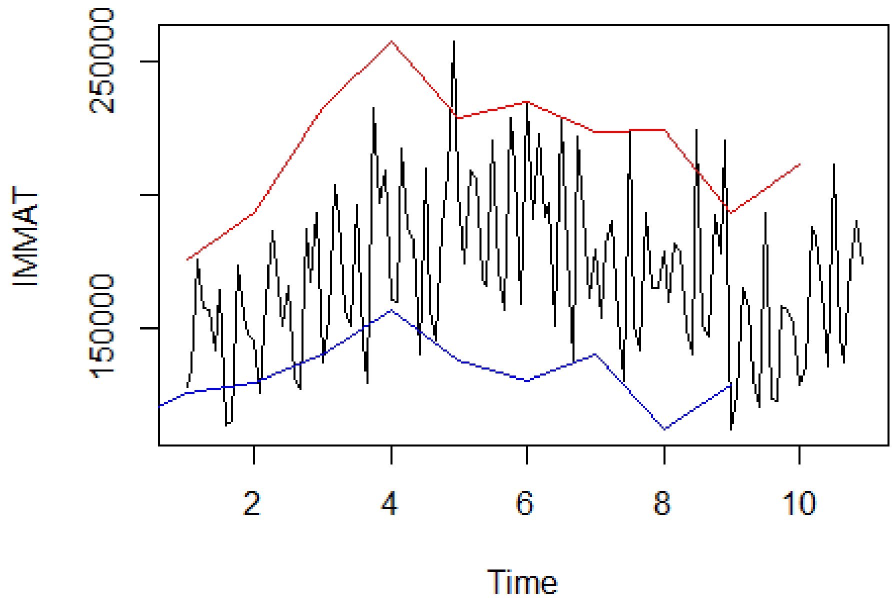


D’après le log out on remarque qu’il y a une tendance peu importante chaque année.

DECOMPOSITION DE LA SERIE TEMPORELLE :

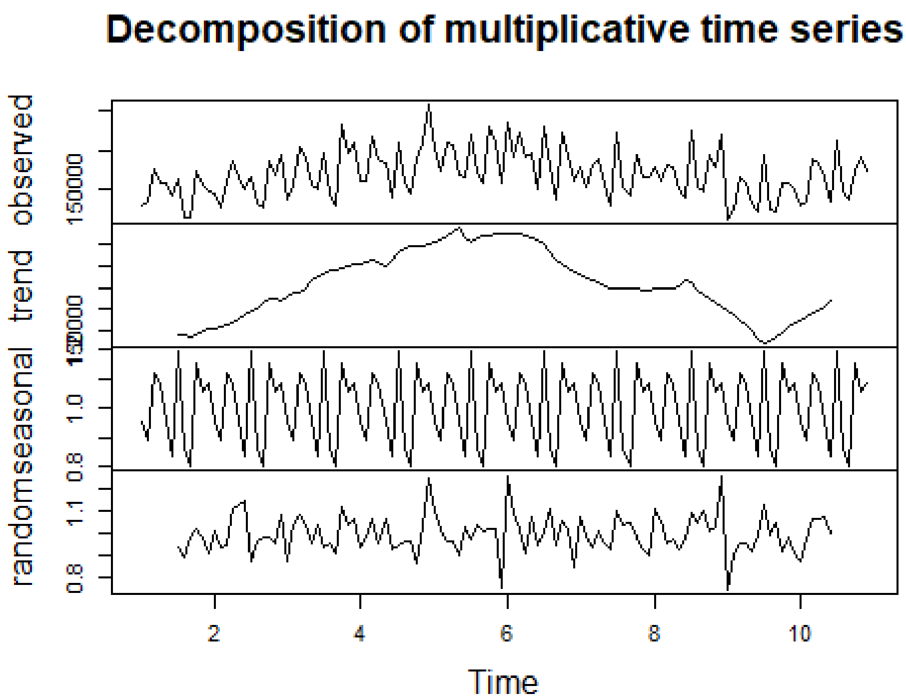
Je commence par faire le test de la bande pour savoir le type de modèle, soit additif ou multiplicatif:

```
MatX=matrix(data=Y,nrow=12)
Min=apply(MatX,2,min)
Max=apply(MatX,2,max)
AnneeMin=c(0:9)
AnneeMax=c(1:10)
plot.ts(Y)
points(AnneeMin,Min,col="blue",type = "l")
points(AnneeMax,Max,col="red",type = "l")
```



Les deux bandes ne sont pas parallèles, donc le modèle est multiplicatif, maintenant on utilise la fonction decompose:

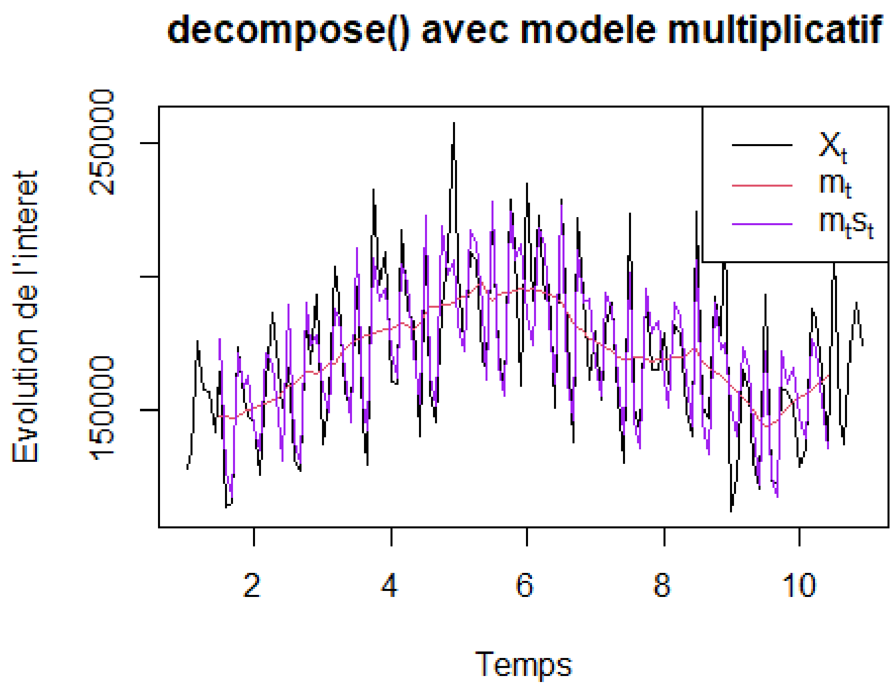
```
fit1 <- decompose(Y,type="multiplicative")
plot(fit1)
```



La distribution des résidus ne semble pas dépendre du temps, ce qui semble indiquer que ce modèle est mieux adapté pour cette série.

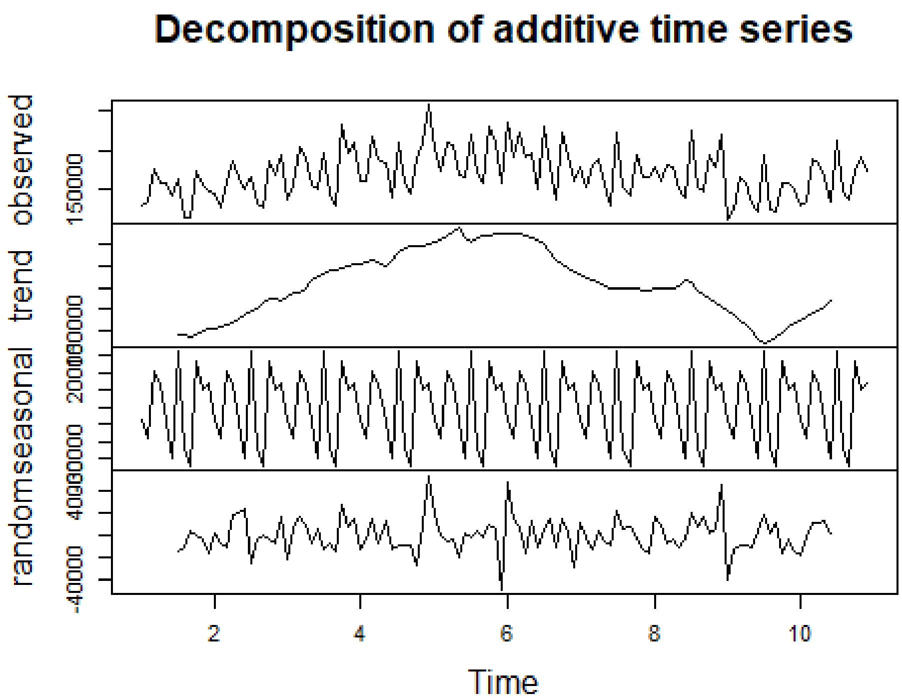
Voyons les prédictions:

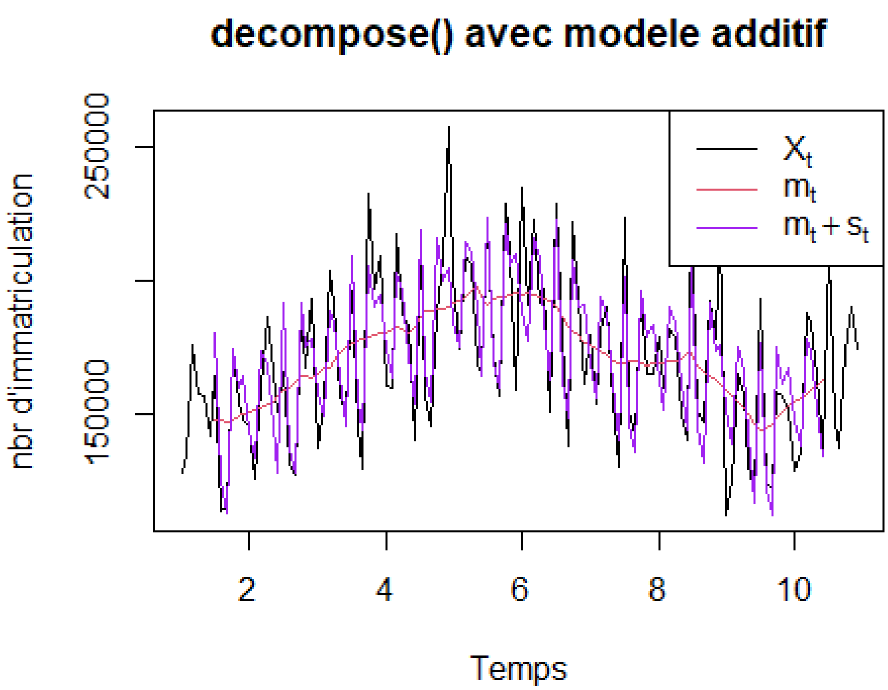
```
plot(Y,xlab="Temps",ylab="Evolution de l’interet",
main="decompose() avec modele multiplicatif")
points(fit1$trend,type="l",col=2)
points(fit1$trend*fit1$seasonal,type="l",col="purple")
legend("topright",c(expression(X[t]),expression(m[t]),expression(m[t]*s[t])),
col=c(1,2,"purple"),lty=1)
```



Les prédictions ne sont pas parfaites, il y a des sous-estimations et des surestimations sauf l'année 7 dont il y a une prédiction un peu meilleure.

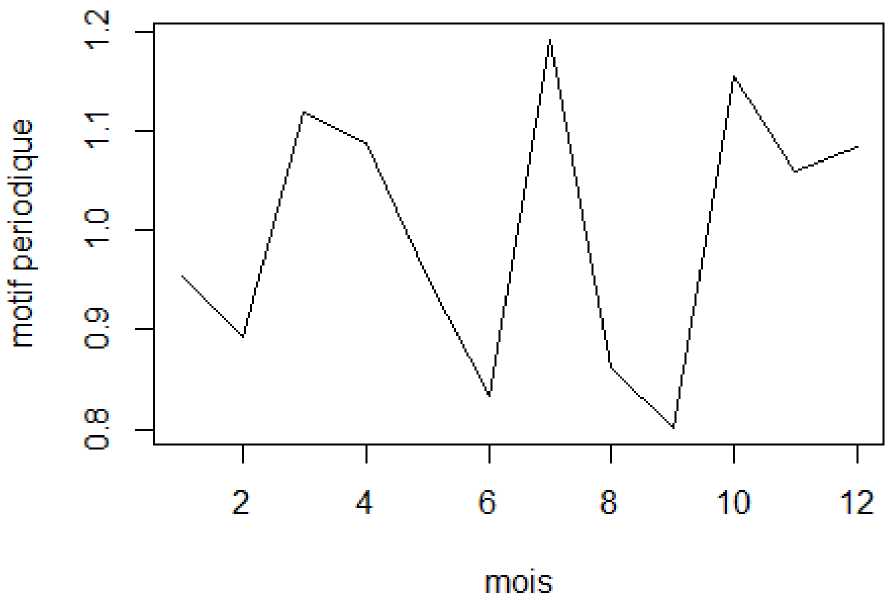
Jetons un œil sur la prédiction du model additif pour voir si il est pertinent :





En comparant les deux modèles, il n'y a pas de grande différence ce qui rend ce model complexe, mais en se basant sur la méthode de la bande, je choisis de travailler avec le model multiplicatif.

```
plot(fit1$figure,type="l",xlab="mois",ylab="motif periodique")
```



Je remarque plusieurs piques maximales en mars, juillet et octobre et deux piques minimales en juin et septembre.

PREDICTION :

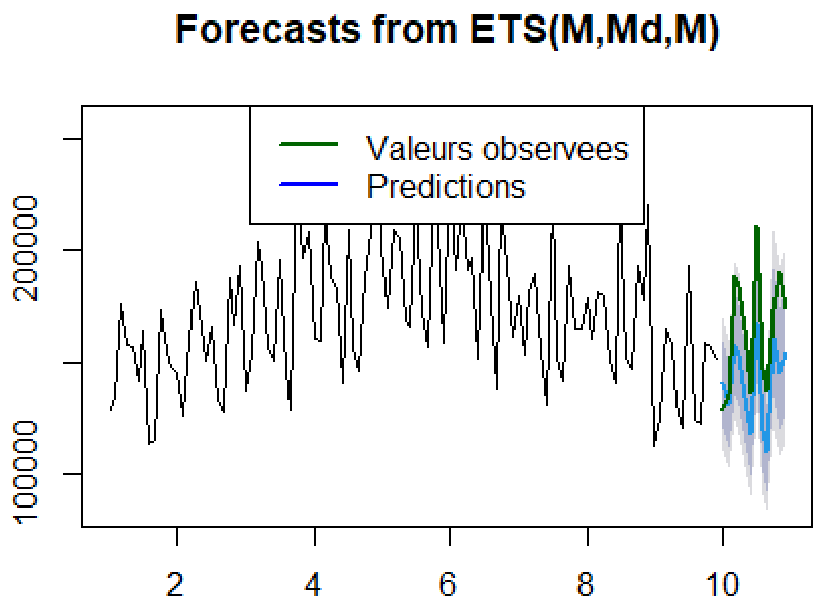
je commence par enlever la dernière année pour la comparer avec mes prédictions:

```
Y.19 <- window(Y,start=1,end=c(9,12))
Y.10 <- window(Y,start=10)
```

LISSAGE EXPONENTIELLE :

Je passe directement au lissage exponentiel triple :

```
fitHW = ets(Y.19,model="MMM")

predHW = forecast(fitHW,h=12)
plot(predHW)
points(Y.10,type="l",col="darkgreen",lwd=2)
legend("top",c("Valeurs observees","Predictions"), col=c("darkgreen","blue"),
lty=rep(1,2),lwd = rep(2,2))
```

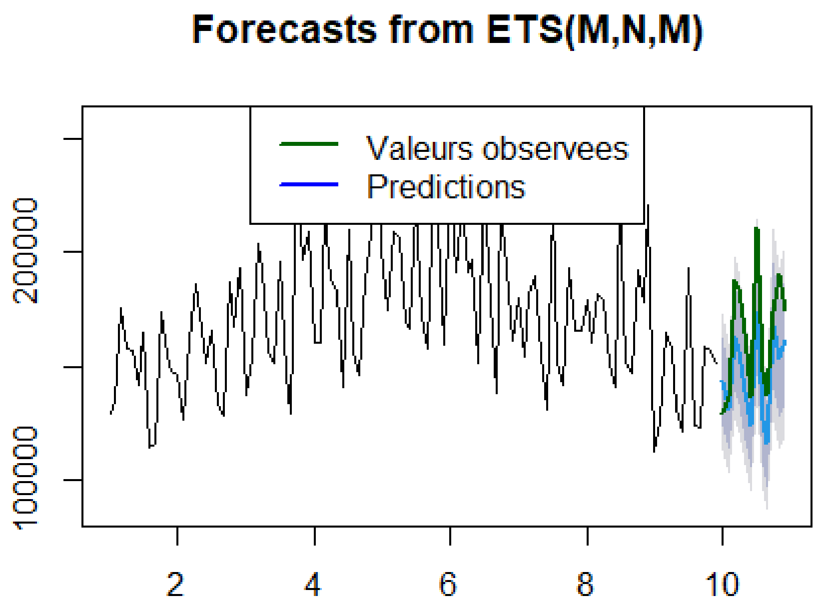



```
predict(fitHW,12)

##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## Jan 10       140614.5 121084.58 160011.2 110996.57 169694.8
## Feb 10       130640.2 112166.18 148843.5 102219.00 158417.8
## Mar 10       158101.8 134467.09 181776.4 122499.13 194619.7
## Apr 10       153445.3 130706.56 176574.4 119111.17 188609.0
## May 10       134906.3 114311.27 156143.5 103662.94 167096.3
## Jun 10       118283.8  99666.95 137777.5  90616.77 149085.4
## Jul 10       168141.6 141627.66 196648.0 127728.76 211512.5
## Aug 10       120530.7 101400.39 140891.9  90983.17 153241.2
## Sep 10       110188.5  92127.23 130163.1  83397.77 140704.1
## Oct 10       161470.3 134025.25 192036.2 120830.34 208470.7
## Nov 10       145790.0 120925.08 173689.4 108842.76 189584.9
## Dec 10       154724.0 128187.26 185007.1 115002.69 203356.6
```

Les prédictions ne sont pas assez pertinente, un peu loin des observations.

J'utilise le choix automatique de la fonction forecast pour choisir le modèle, puis je fais les prédictions :

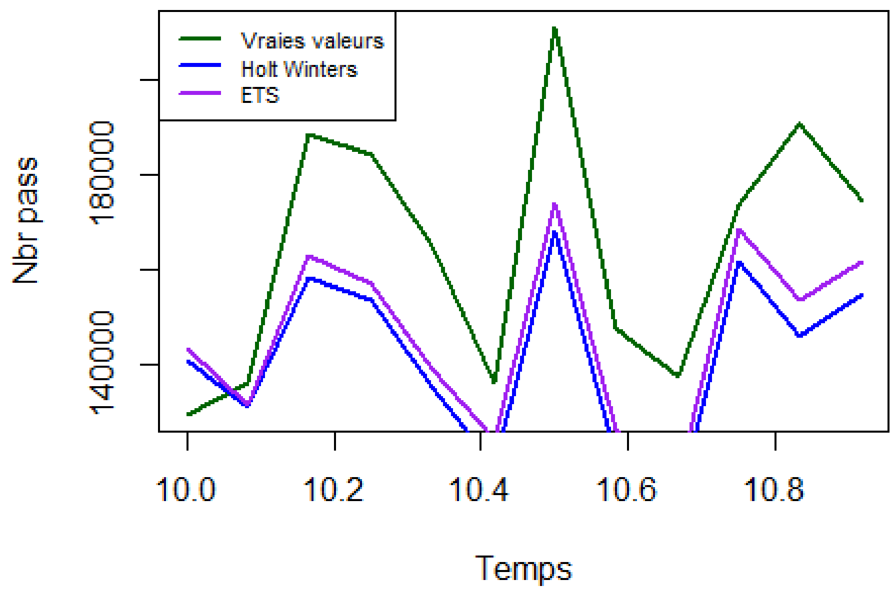
```
fit <- ets(Y.19)
predfit <- forecast(fit,h=12)
plot(predfit)
points(Y.10,type="l",col="darkgreen",lwd=2)
legend("top",c("Valeurs observees","Predictions"), col=c("darkgreen","blue"),
lty=rep(1,2),lwd = rep(2,2))
```




```
summary(fit)

## ETS(M,N,M)
##
## Call:
##  ets(y = Y.19)
##
##   Smoothing parameters:
##     alpha = 0.2225
##     gamma = 1e-04
##
##   Initial states:
##     l = 166864.4468
##     s = 1.1038 1.0482 1.1499 0.7895 0.8621 1.188
##            0.8472 0.9492 1.0721 1.112 0.8979 0.98
##
##   sigma:  0.108
##
##      AIC     AICc      BIC
## 2640.738 2645.956 2680.970
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE      MASE
ACF1
## Training set -885.4727 18141.74 13108.65 -1.526554 7.81743 0.6585737 -0.01
40961
```

Comparison des deux predictions:

```
plot(Y.10,col="darkgreen",lwd=2,ylab="Nbr pass",xlab="Temps")
points(predHW$mean,col="blue",lwd=2,type="l")
points(predfit$mean,col="purple",lwd=2,type="l")
legend("topleft",c("Vraies valeurs","Holt Winters","ETS"),
         col=c("darkgreen","blue","purple"),lty=rep(1,),lwd=rep(2,3),cex=0.7)
```



Je constate que la prédiction faite par default est plus proche des vraies valeurs. Mais il n'y a pas de grandes différences entre cella et le lissage exponentiel. Les deux courbes sont très proches.

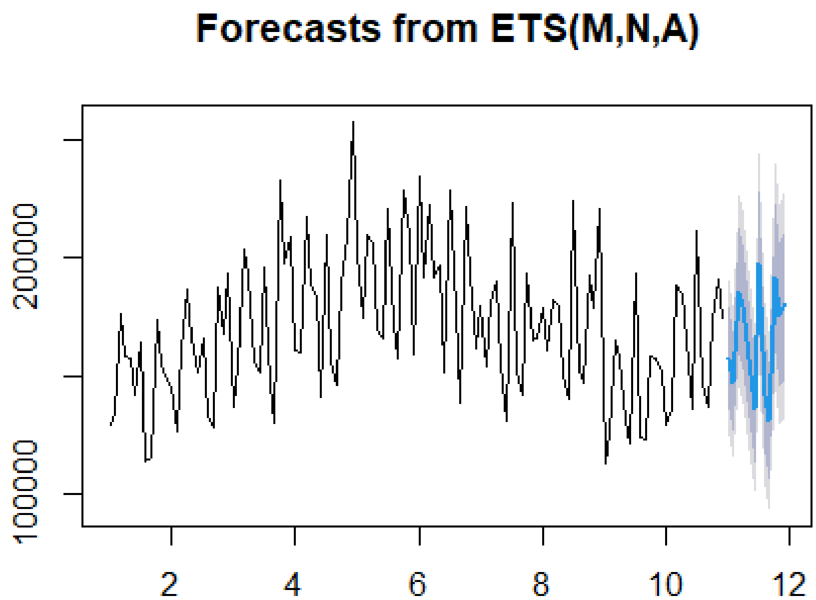
Comparant leur AIC:

```
fit$aic

## [1] 2640.738

fitHW$aic

## [1] 2643.054
```

En se basant sur le AIC, le deuxième model est meilleur que celui du lissage exponentielle. Donc j'utilise ce model par la suite.

PREDICTION DE L'ANNEE D'APRES :

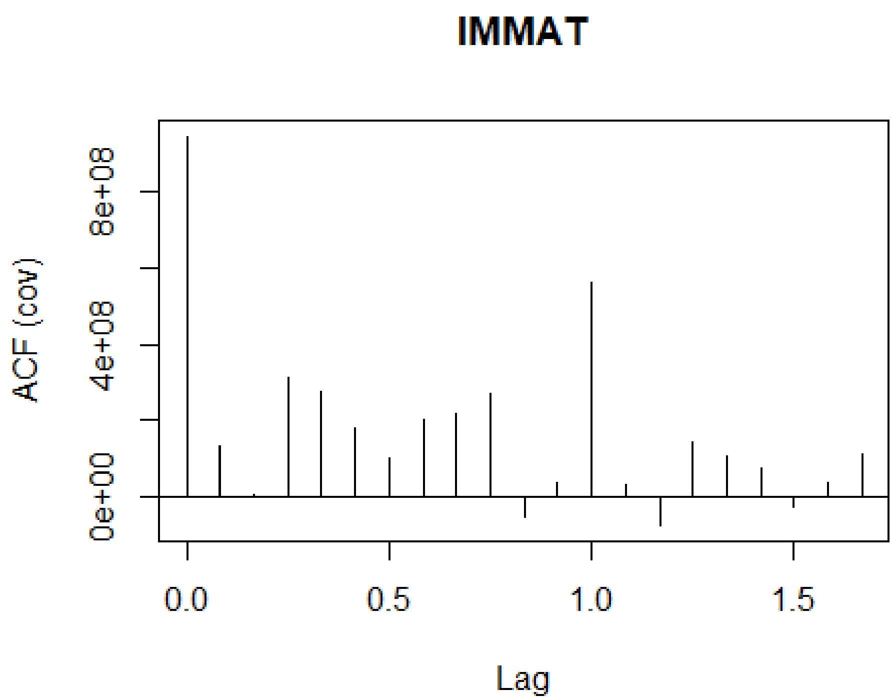
```
fittotal <- ets(Y)

predfittotal <- forecast(fittotal,h=12)
plot(predfittotal)
```



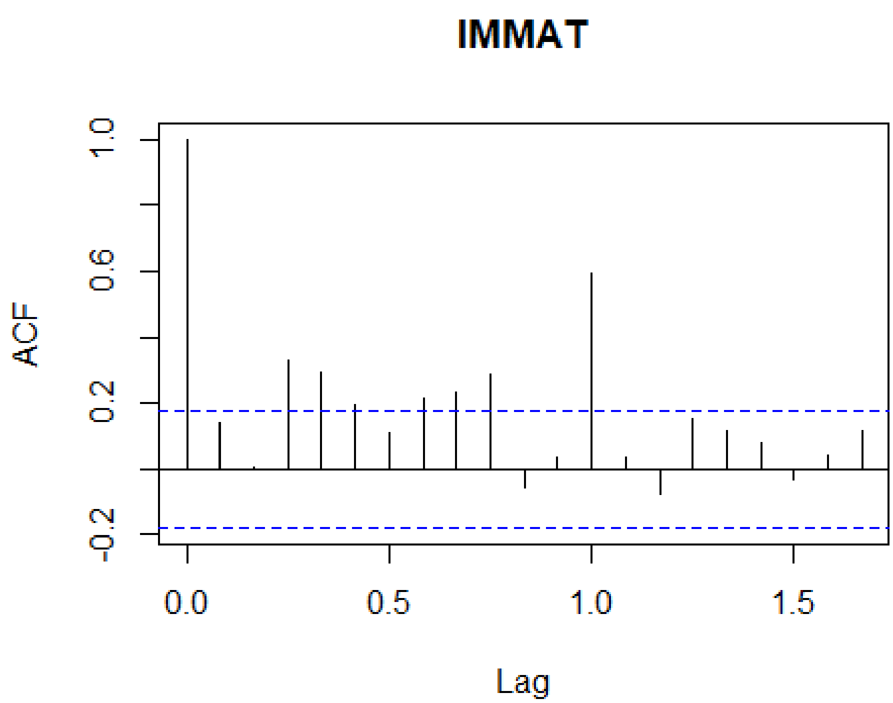
MODELISATION

Estimation de la moyenne et des fonctions d’autocovariance et d’autocorrélation:

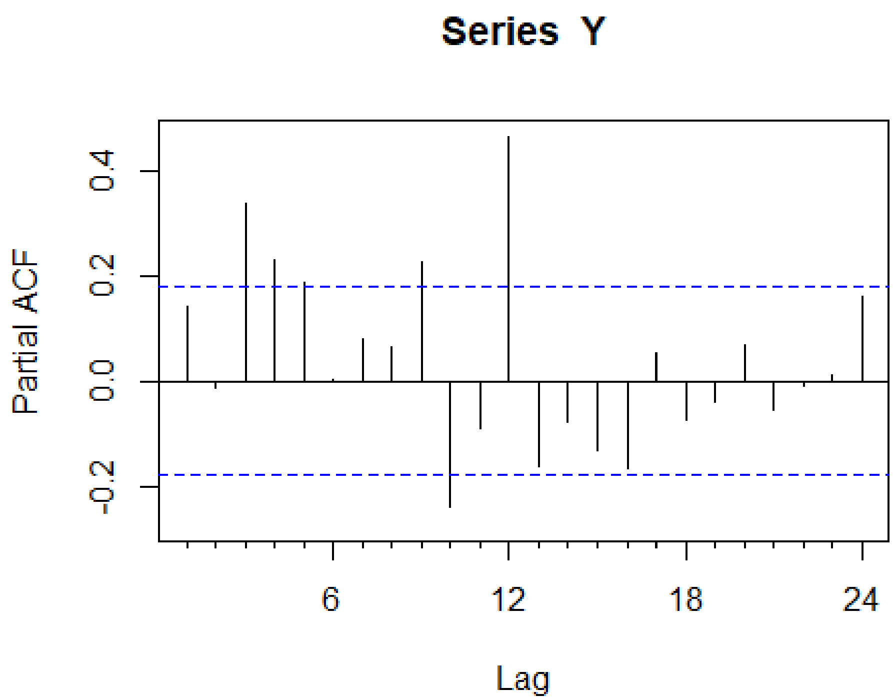
```
mean(Y)

## [1] 170122.8

acf(Y,type ="covariance")
```




```
acf(Y,type ="correlation")
```




```
Pacf(Y)
```



D’après le ACF et le PACF, on n’a pas besoin de faire une transformation, car il y a une décroissance exponentielle vers 0.

Faisons le Box.test pour vérifier la blancheur du résidus:

```
length(Y)

## [1] 120

Box.test(Y,lag=20,type="Box-Pierce")

##
##  Box-Pierce test
##
## data:  Y
## X-squared = 104.45, df = 20, p-value = 1.995e-13
```

La P_valeur est plus petite que 5%, donc la blancheur n’est pas vérifiée.

la fonction auto.arima nous donne le modèle le plus convenable pour nos données:

```
auto.arima(Y.19)

## Series: Y.19
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##             ma1      sma1
```

```
##       -0.8042  -0.6678
## s.e.   0.0564   0.1345
##
## sigma^2 = 438328329:  log likelihood = -1083.08
## AIC=2172.15   AICc=2172.41   BIC=2179.81
```

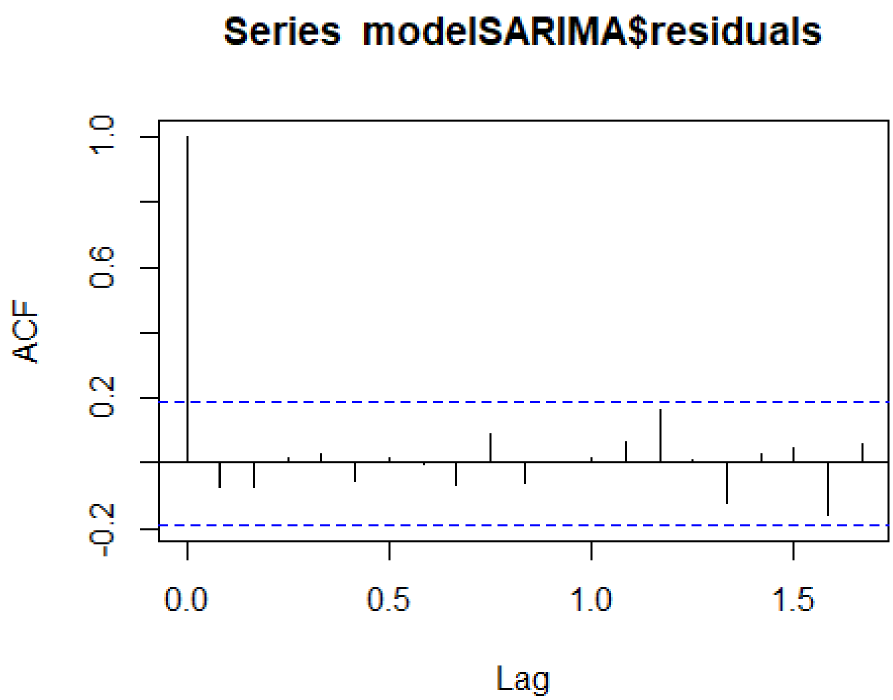
Donc notre modèle sera une SARIMA:

```
modelSARIMA=auto.arima(Y.19)
modelSARIMA

## Series: Y.19
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##           ma1     sma1
##       -0.8042  -0.6678
## s.e.   0.0564   0.1345
##
## sigma^2 = 438328329:  log likelihood = -1083.08
## AIC=2172.15   AICc=2172.41   BIC=2179.81

t_stat(modelSARIMA)

##             ma1      sma1
## t.stat -14.2695 -4.963611
## p.val    0.0000  0.000001

cor.arma(modelSARIMA)

##             ma1       sma1
## ma1   1.0000000 -0.3036196
## sma1 -0.3036196  1.0000000

Box.test(modelSARIMA$residuals,lag=20)

##
##  Box-Pierce test
##
## data:  modelSARIMA$residuals
## X-squared = 11.64, df = 20, p-value = 0.9279
```

La P_valeur est plus grande que 5%, donc la blancheur des résidus est vérifié. La corrélations entre les variables est inférieure à 0.9, donc le model est parfait.

Vérifiant le ACF des résidus de ce model :

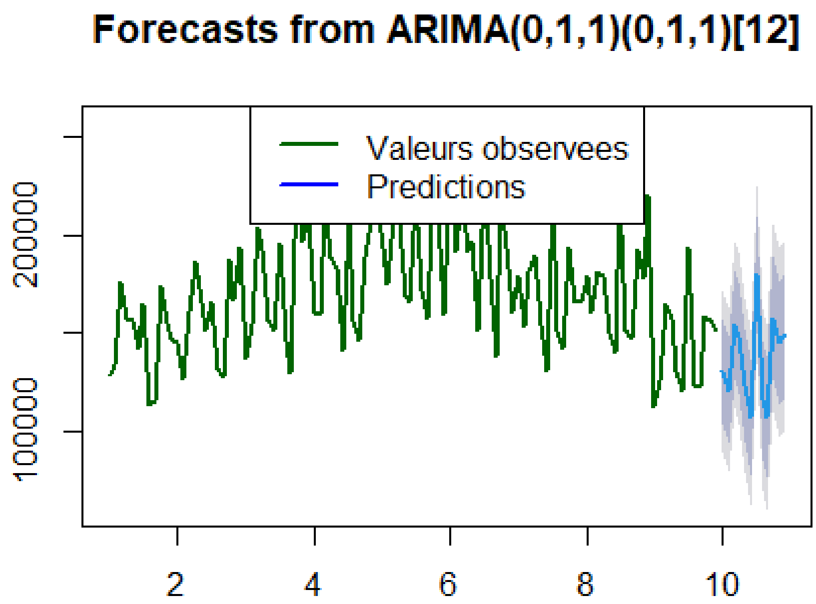
```
acf(modelSARIMA$residuals)
```



Comme on peut voir, l'ACF est bien.

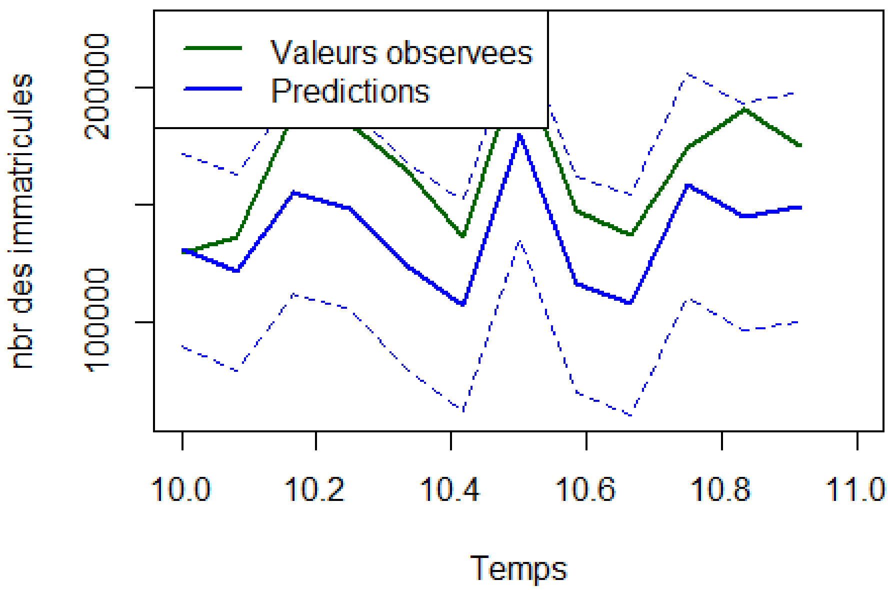
PREDICTION

```
predSARIMA=forecast(modelSARIMA,12)
predSARIMA

##        Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## Jan 10       130658.3 103819.28 157497.3  89611.57 171705.0
## Feb 10       121010.7  93662.07 148359.3  79184.60 162836.7
## Mar 10       154585.4 126736.58 182434.3 111994.29 197176.6
## Apr 10       148089.3 119748.97 176429.5 104746.53 191432.0
## May 10       123700.5  94877.18 152523.9  79619.03 167782.0
## Jun 10       106961.5  77663.10 136260.0  62153.45 151769.6
## Jul 10       180298.1 150532.13 210064.0 134774.99 225821.1
## Aug 10       116114.2  85887.95 146340.4  69887.15 162341.2
## Sep 10       107151.7  76472.10 137831.3  60231.30 154072.1
## Oct 10       158093.8 126967.42 189220.2 110490.12 205697.5
## Nov 10       144999.1 113432.32 176566.0  96721.86 193276.4
## Dec 10       149043.7 117042.49 181044.9 100102.08 197985.3

plot(predSARIMA)
points(Y.19,type="l",col="darkgreen",lwd=2)
legend("top",c("Valeurs observees","Predictions"),col=c("darkgreen","blue"),
 lty=rep(1,2),lwd = rep(2,2))
```



Comparons Les prédictions avec les vrais valeurs :

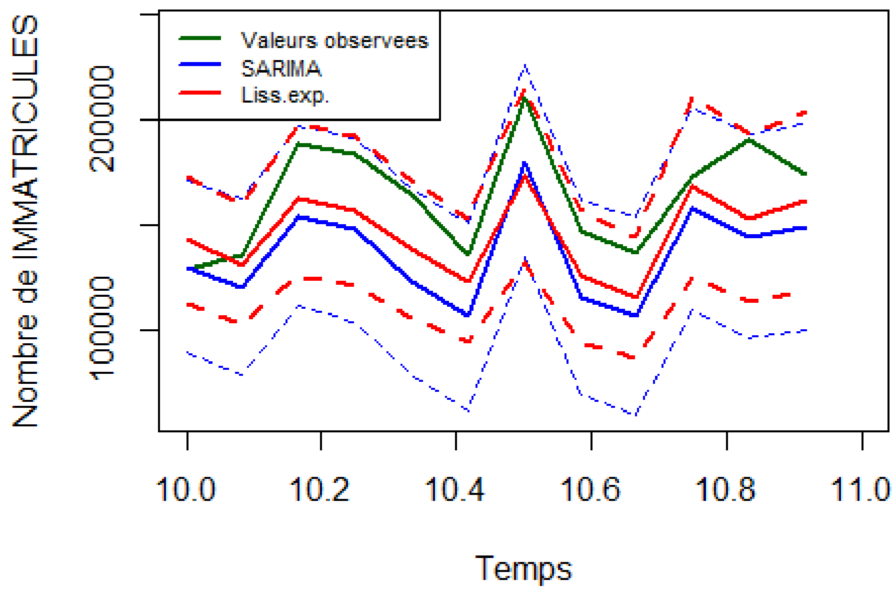
```
plot(Y.10,col="darkgreen",lwd=2,ylab="nbr des immatricules",xlab="Temps",xlim
=c(10,11),
 ylim=range(c(Y.10,predSARIMA$lower,predSARIMA$upper)))
points(predSARIMA$mean,col="blue",lwd=2,type="l")
points(predSARIMA$lower[,2],col="blue",type="l",lty=2)
points(predSARIMA$upper[,2],col="blue",type="l",lty=2)
legend("topleft",c("Valeurs observees","Predictions"),
col=c("darkgreen","blue"),lty=rep(1,2),lwd = rep(2,2))
```



Le modèle n’est pas parfait, mais les deux courbes sont proche et donc notre modèle est assez bon.

Faisons une dernière comparaison entre SARIMA et le modèle choisi auparavant “predfittotal”:

```
fittotal <- ets(Y)
predfittotal <- forecast(fittotal,h=12)

plot(Y.10,col="darkgreen",lwd=2,ylab="Nombre de IMMATRICULES",xlab="Temps",
xlim=c(10,11),ylim=range(c(Y.10,
predSARIMA$lower,predSARIMA$upper,predfittotal$lower,predfittotal$upper)))
points(predSARIMA$mean,col="blue",lwd=2,type="l")
points(predSARIMA$lower[,2],col="blue",type="l",lty=2)
points(predSARIMA$upper[,2],col="blue",type="l",lty=2)
points(predfittotal$mean,col="red",lwd=2,type="l")
points(predfittotal$lower[,2],col="red",lwd=2,type="l",lty=2)
points(predfittotal$upper[,2],col="red",lwd=2,type="l",lty=2)
legend("topleft",c("Valeurs observees","SARIMA", "Liss.exp."),
 col=c("darkgreen","blue","red"),lty=rep(1,3),lwd = rep(2,3),cex=0.7)
```



Conclusion :

SARIMA et Lissage Exponentiel sont relativement pareil, mais je prendrais le Lissage Exponentiel qui s'approche plus des valeurs réelles avec un intervalle de confiance moins large.