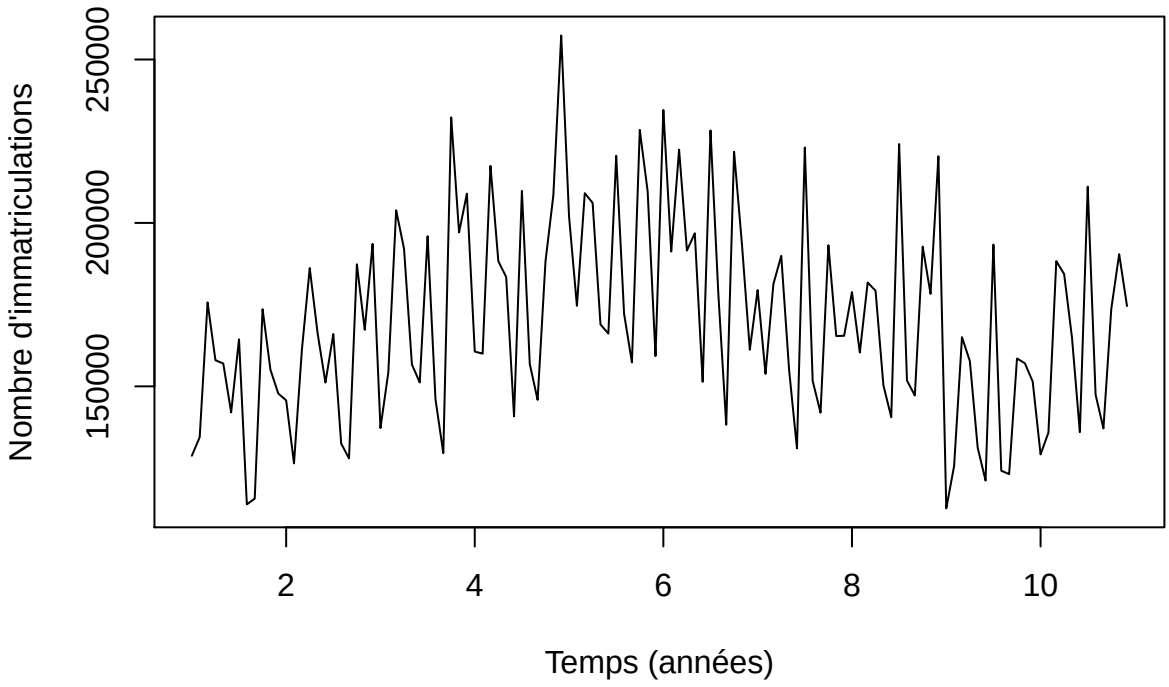
TP4 : prévision

Elements de correction

Dans le sujet, les questions en rouge sont des questions type examen.

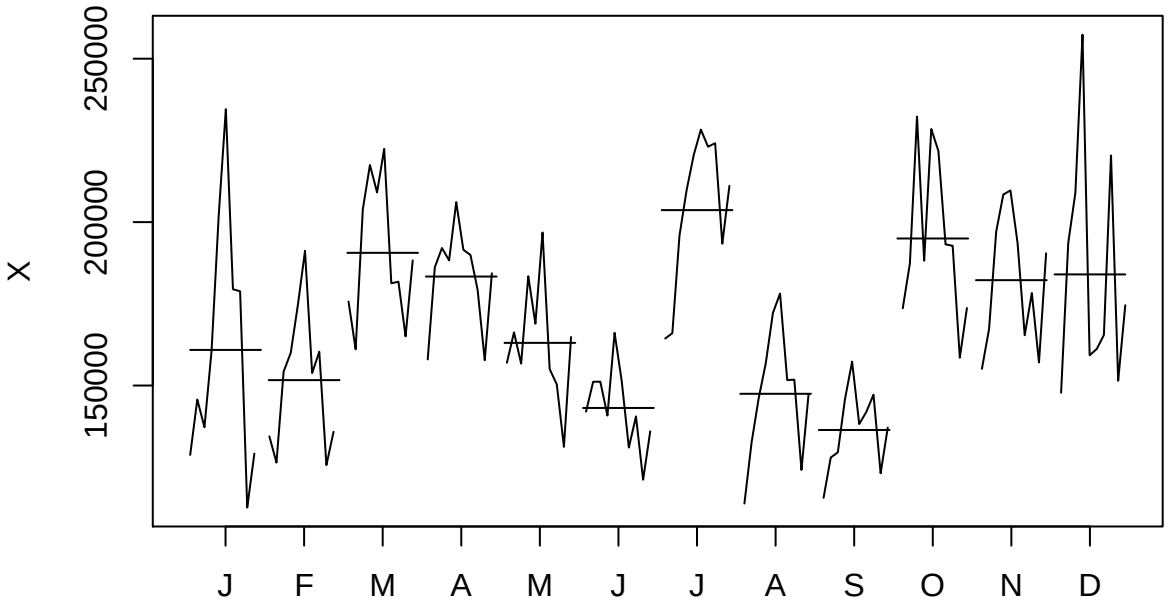
L'objectif de ce TP est de comparer différentes méthodes de prévision vues en cours. Nous nous intéressons à l'évolution sur 10 ans du nombre d'immatriculations de voitures particulières en France.

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

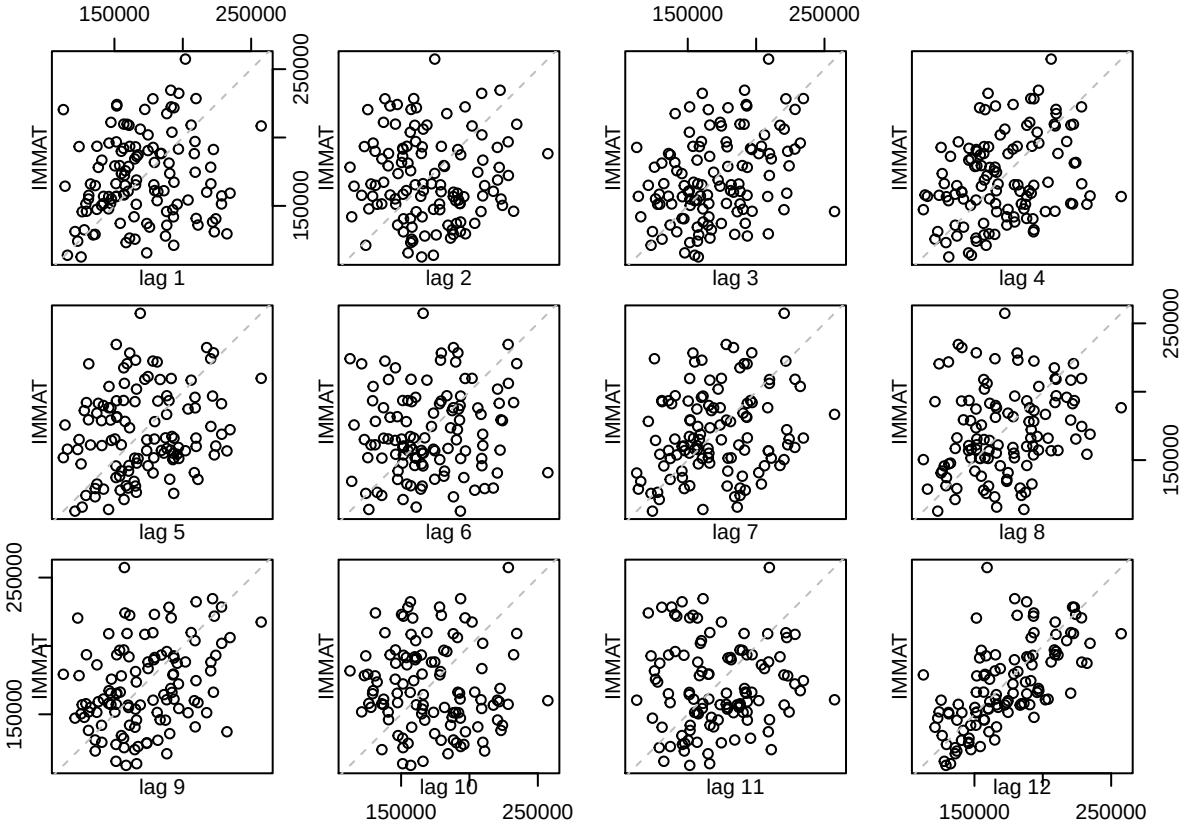


Nous allons dans un premier temps étudier la série d'un point de vue graphique.

```
monthplot(X)
```




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



Nous observons de fortes variations saisonnières et une autocorrélation forte au lag 12, ce qui nous oriente vers un modèle saisonnier.

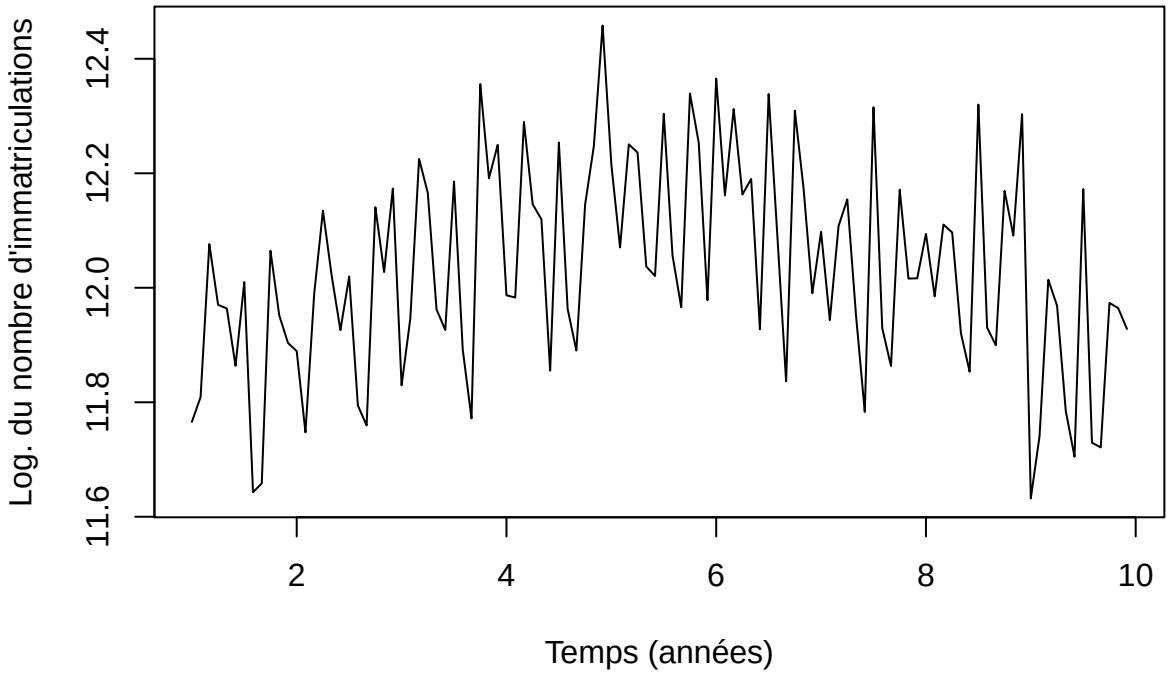
Pour évaluer les qualités prédictives des modèles considérés, nous séparons la dernière année des neuf précédentes.

```
X.app <- window(X,end=c(9,12))
X.test <- window(X,start=10)
```

Commenter les représentations graphiques de la série (chronogramme, monthplot et lagplot).

Pour stabiliser la variance nous étudierons la série transformée $Y_t = \log(X_t)$,

```
Y.app <- log(X.app)
Y.test <- log(X.test)
plot(Y.app,ylab="Log. du nombre d'immatriculations",xlab="Temps (années)")
```



Construction d'un modèle SARIMA

Tests de stationnarité

Nous faisons tout d'abord un test de Dickey-Fuller augmenté. Comme la série montre une tendance, nous choisissons l'option `type='trend'`. Pour choisir p, nous commençons par choisir une valeur élevée ($p = 6$) et nous diminuons cette valeur jusqu'à ce que le coefficient correspondant au plus grand retard soit significatif. Nous choisissons $p = 4$.

```
library(urca)
testDF6 <- ur.df(Y.app,lags=6,type='trend')
summary(testDF6)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3766 -0.1237  0.0078  0.1021  0.4220
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.2114290  2.0857290   1.540 0.127062
## z.lag.1     -0.2644638  0.1735532  -1.524 0.130984
## tt          -0.0005191  0.0005940  -0.874 0.384490
## z.diff.lag1 -0.6987969  0.1846010  -3.785 0.000273 ***
## z.diff.lag2 -0.8434372  0.1942564  -4.342 3.63e-05 ***
## z.diff.lag3 -0.5462690  0.2013275  -2.713 0.007952 **
## z.diff.lag4 -0.3870695  0.1883317  -2.055 0.042689 *
## z.diff.lag5 -0.1591610  0.1473605  -1.080 0.282930
## z.diff.lag6 -0.1446722  0.1068163  -1.354 0.178925
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1696 on 92 degrees of freedom
## Multiple R-squared:  0.5447, Adjusted R-squared:  0.5052
## F-statistic: 13.76 on 8 and 92 DF,  p-value: 5.983e-13
##
##
## Value of test-statistic is: -1.5238 1.1696 1.7436
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

```
testDF4 <- ur.df(Y.app,lags=4,type='trend')
summary(testDF4)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39787 -0.10199  0.01107  0.09109  0.44003
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.9008048  1.9277941   2.023 0.045806 *
## z.lag.1     -0.3226290  0.1603721  -2.012 0.047049 *
## tt          -0.0003357  0.0005639  -0.595 0.553035
```

```
## z.diff.lag1 -0.6362775  0.1654489  -3.846 0.000216 ***
## z.diff.lag2 -0.7475026  0.1600639  -4.670 9.79e-06 ***
## z.diff.lag3 -0.4193040  0.1369682  -3.061 0.002858 **
## z.diff.lag4 -0.2136584  0.1014317  -2.106 0.037776 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.168 on 96 degrees of freedom
## Multiple R-squared:  0.5361, Adjusted R-squared:  0.5071
## F-statistic: 18.49 on 6 and 96 DF,  p-value: 3.56e-14
##
##
## Value of test-statistic is: -2.0118 1.5668 2.3436
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

La ligne `Value of the test statistic is` nous donne, dans l'ordre et avec les notations du cours :

- la valeur de la statistique du test $H_0 : (\beta_1, \pi) = (0, 0)$ (non stationnaire) contre $H_1 : \pi < 0$. On rejete H_0 lorsque la valeur de cette statistique est **inférieure** à la valeur critique au niveau considéré (première ligne du tableau `Critical values for test statistics`),
- la valeur de la statistique du test $H_0 : (\beta_1, \beta_2, \pi) = (0, 0, 0)$ contre $H_1 : (\beta_1, \beta_2, \pi) \neq (0, 0, 0)$. On rejete H_0 lorsque la valeur de cette statistique est **supérieure** à la valeur critique au niveau considéré (deuxième ligne du tableau `Critical values for test statistics`),
- la valeur de la statistique du test $H_0 : (\beta_2, \pi) = (0, 0)$ contre $H_1 : (\beta_2, \pi) \neq (0, 0)$. On rejete H_0 lorsque la valeur de cette statistique est **supérieure** à la valeur critique au niveau considéré (deuxième ligne du tableau `Critical values for test statistics`).

Interprétez les résultats des tests de Dickey-Fuller.

Le test conclu à l'acceptation des hypothèses H_0 pour les trois tests, ce qui indique que les données ne sont pas en contradiction avec la présence d'une racine unitaire.

Il est difficile de déterminer à partir de l'examen du chronogramme de la série si $\beta_2 = 0$ ou $\beta_2 \neq 0$ car la présence d'une tendance peut-être due uniquement au processus non stationnaire $(R_t)_{t\in\mathbb{Z}}$. Nous faisons donc les deux tests.

```
testKPSStau <- ur.kpss(Y.app,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
```

Nous rejettons $H_0 : (X_t)_{t\in\mathbb{Z}}$ *stationnaire* lorsque la statistique de test dépasse les valeurs critiques au niveau

considéré, ce qui est le cas ici. Nous considérons donc que le processus n'est pas stationnaire.

```
testKPSSmu <- ur.kpss(Y.app,type='mu')
summary(testKPSSmu)
```

```
##
## ######################
## # KPSS Unit Root Test #
## ######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.457
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Nous rejettons H_0 : $(X_t)_{t\in\mathbb{Z}}$ *stationnaire* ***à une tendance linéaire déterministe près*** lorsque la statistique de test dépasse les valeurs critiques au niveau considéré, ce qui n'est pas le cas ici pour le niveau 5% (au vu des valeurs critiques et de la valeur de la statistique de test, la p-valeur du test est comprise entre 5% et 10%).

Au vu des résultats des tests, nous pouvons penser raisonnablement que le processus est de type :

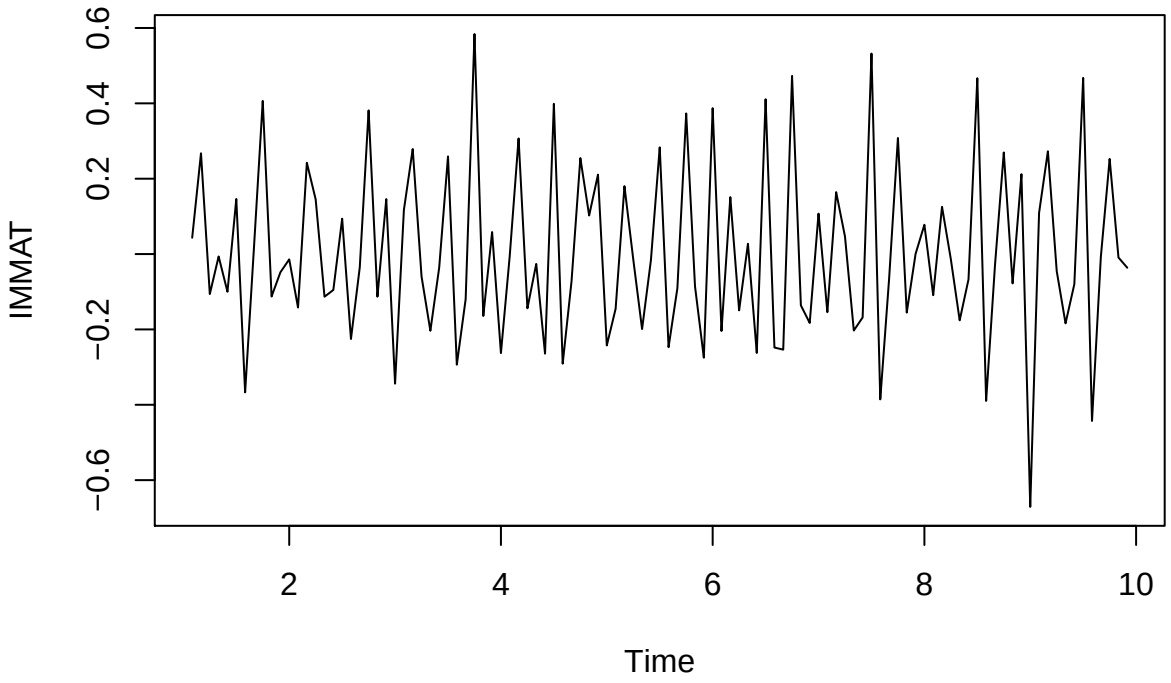
$$X_t = \beta_1 + R_t + U_t,$$

avec $\beta_1 \in \mathbb{R}$, $(R_t)_{t\in\mathbb{Z}}$ non stationnaire tel que $R_t = R_{t-1} + Z_t$ avec $\{Z_t\}_{t\in\mathbb{Z}}$ un bruit blanc (nous disons que $(R_t)_{t\in\mathbb{Z}}$ est une *marche aléatoire*) et $(U_t)_{t\in\mathbb{Z}}$ un processus stationnaire, cela nous oriente donc vers un processus de type ARIMA.

Etude de la série différenciée

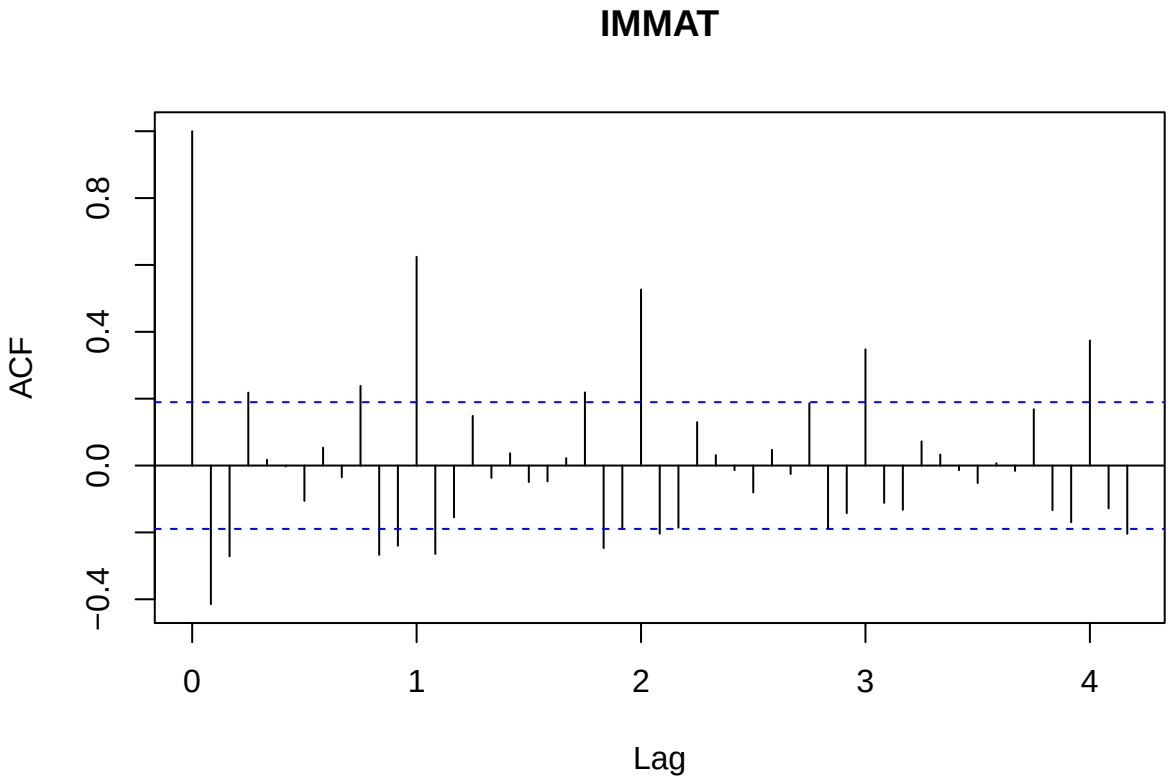
Nous différencions donc une fois la série (ce qui permet d'éliminer la partie marche aléatoire):

```
DeltaY.app=diff(Y.app)
plot(DeltaY.app)
```

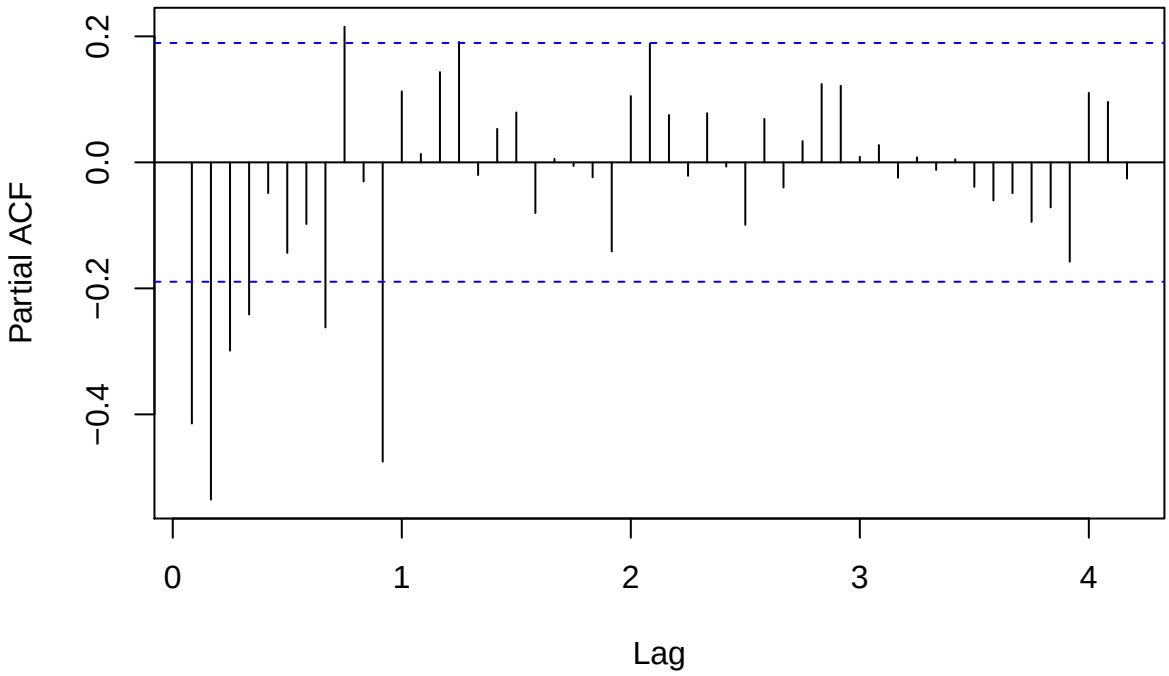


La série différenciée ne semble pas avoir de tendance même si la variance semble augmenter avec le temps. Traçons les fonctions d'autocorrélation et d'autocorrélation partielle.

```
acf(DeltaY.app,lag.max=50)
```




```
pacf(DeltaY.app,lag.max=50)
```



Au vu du tracé des fonctions d'auto-corrélation et d'autocorrélation partielle : quelle modélisation pouvons-nous proposer pour la série 'Y.app' ?

La fonction d'autocorrélation présente des pics significatifs pour le premier retard et pour les retards aux pas multiples de 12. La fonction d'autocorrélation partielle une décroissance sinusoïdale amortie. Nous pouvons donc penser à un modèle du type $SARIMA(0,1,q)(0,D,Q)_{12}$. Pour rappel, $(X_t)_{t\in\mathbb{Z}}$ suit un mdèle de type $SARIMA(p,d,q)(P,D,Q)_s$ s'il existe des polynômes Φ et Φ_s de degrés respectifs p et P n'ayant pas de racine égale à 1 et des polynômes Θ et Θ_s de degrés respectifs q et Q tels que :

$$(I-B)^d\Phi(B)X = \Theta(B)U, \tag{1}$$

et

$$(I-B^s)^D\Phi_s(B^s)U = \Theta_s(B^s)Z, \tag{2}$$

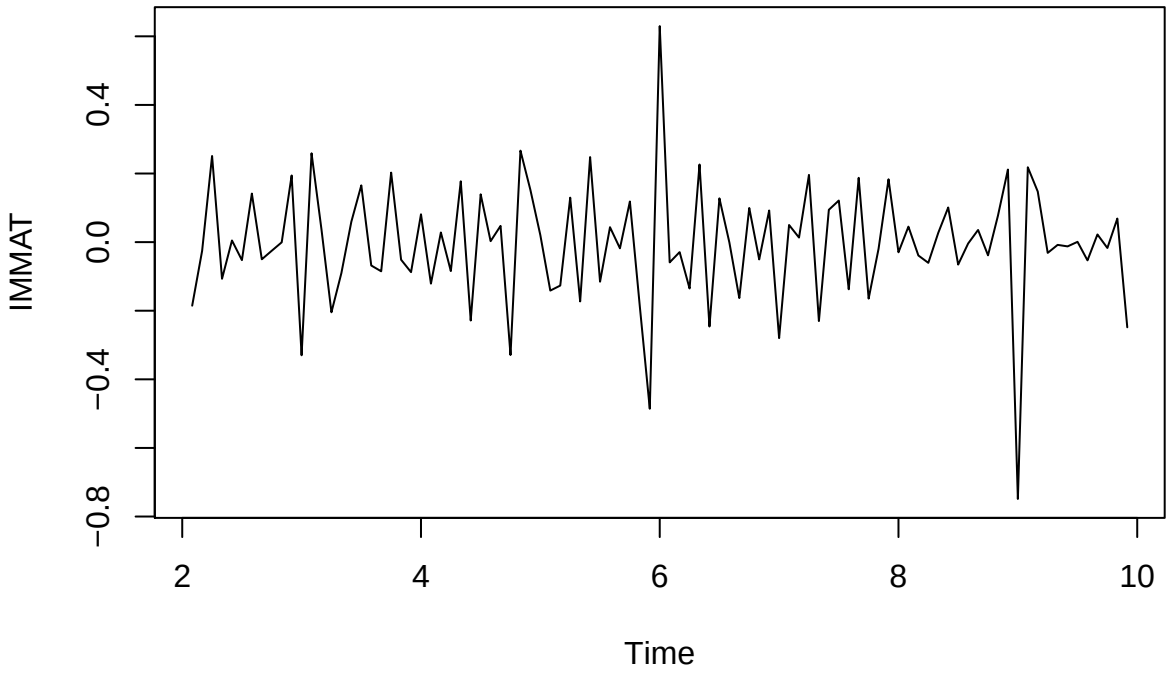
avec Z un bruit blanc.

Les équations (1) et (2) peuvent être combinées en une seule équation :

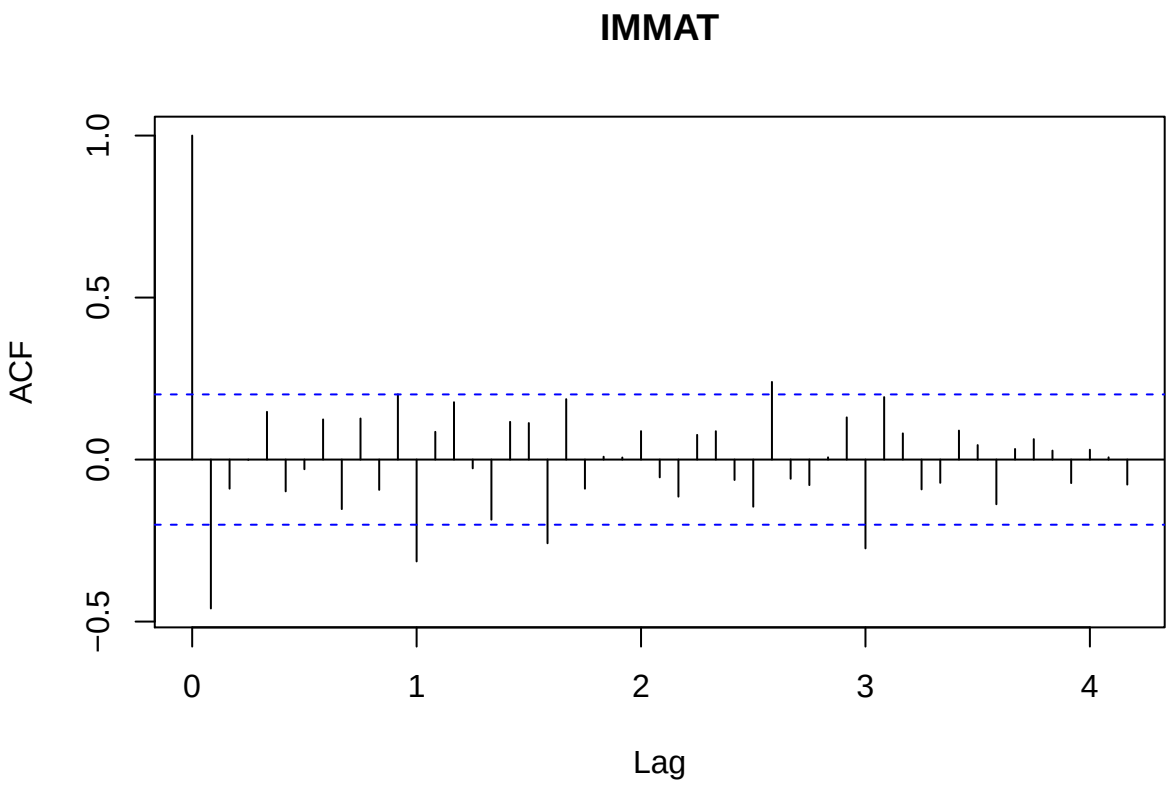
$$(I-B^s)^D(I-B)^d\Phi(B)\Phi_s(B^s)X = \Theta(B)\Theta_s(B^s)Z. \tag{3}$$

Il reste maintenant à identifier q, D et Q. Comme les pics aux retards multiples de 12 ne semblent pas décroître exponentiellement, nous supposons que $D \neq 0$ (c'est-à-dire que nous supposons que le processus $(U_t)_{t\in\mathbb{Z}}$ des équations (1) et (2) n'est pas stationnaire). Pour éviter de considérer des modèles trop complexes, il est généralement conseillé de satisfaire la condition $d+D \leq 2$ (c'est-à-dire nous restreindre ici au cas $D=1$). Nous allons faire une différentiation saisonnière pour supprimer le terme $(I-B^s)$ de l'équation (3). En effet, si X vérifie l'équation (3) avec $d=D=1$ et $s=12$, alors le processus $\Delta_{12}\Delta X = (I-B^{12})(I-B)X$ vérifie une équation $SARMA(p,q)(P,Q)_{12}$. Observons donc les fonctions d'autocorrélation et d'autocorrélation partielle de notre processus $\Delta_{12}\Delta Y$:

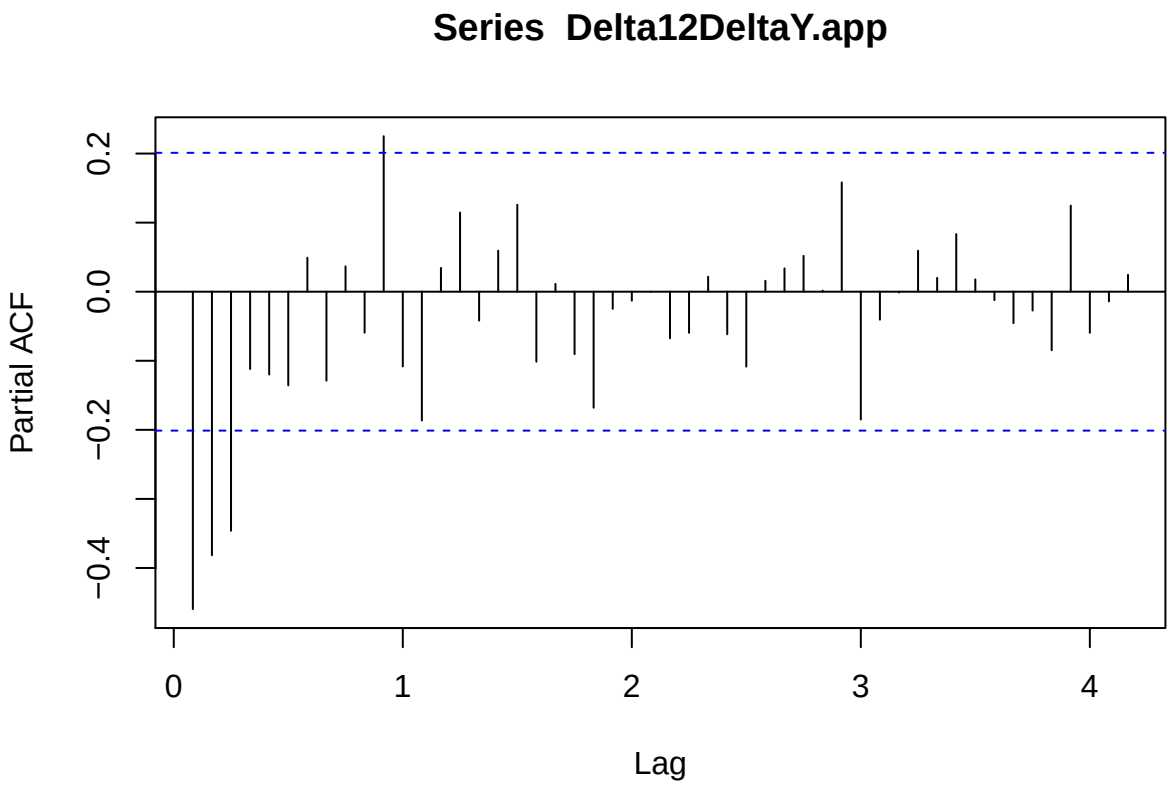
```
Delta12DeltaY.app = diff(DeltaY.app,lag=12)
plot(Delta12DeltaY.app)
```




```
acf(Delta12DeltaY.app,lag.max=50)
```



```
pacf(Delta12DeltaY.app,lag.max=50)
```



La fonction d'autocorrélation montre un pic qui semble significatifs au retard 1 (d'autres pics apparaissent également, notamment aux retards 12 et 36) et une décroissance exponentielle de la fonction d'autocorrélation partielle. Nous commençons par proposer un modèle de type $SARIMA(0,1,1)(0,1,0)_{12}$ et observons les résidus.

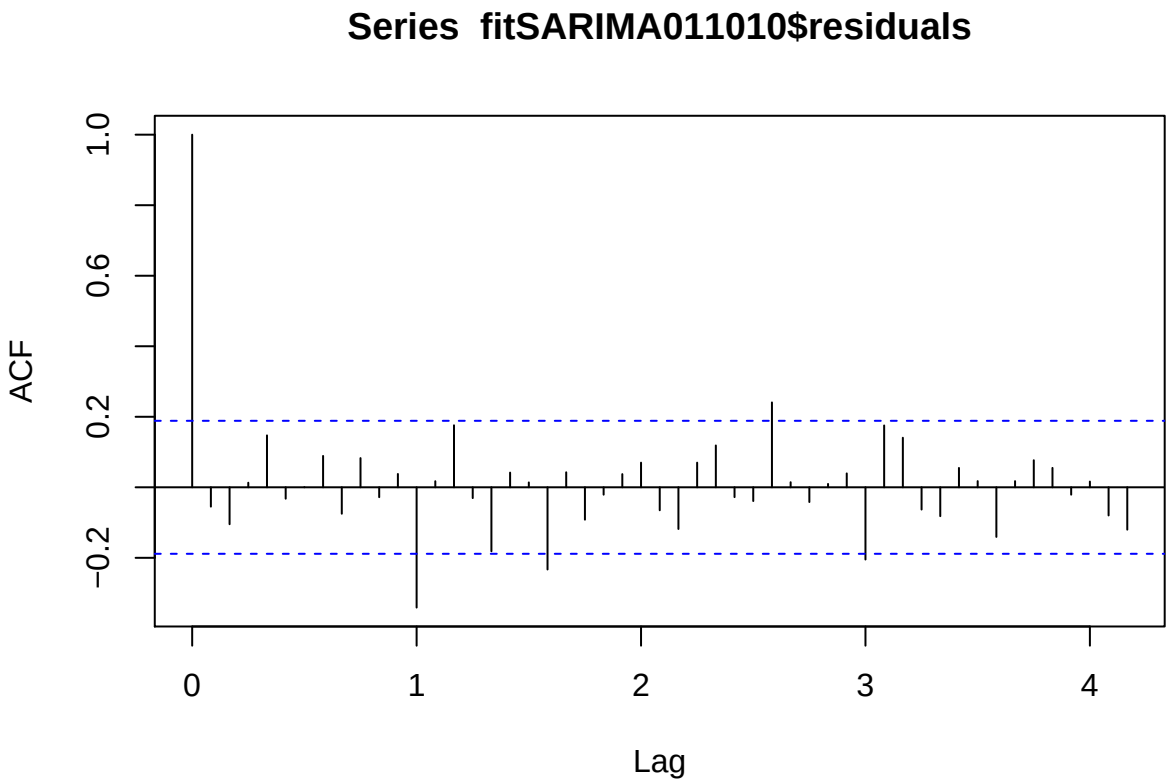
```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.3.2
```

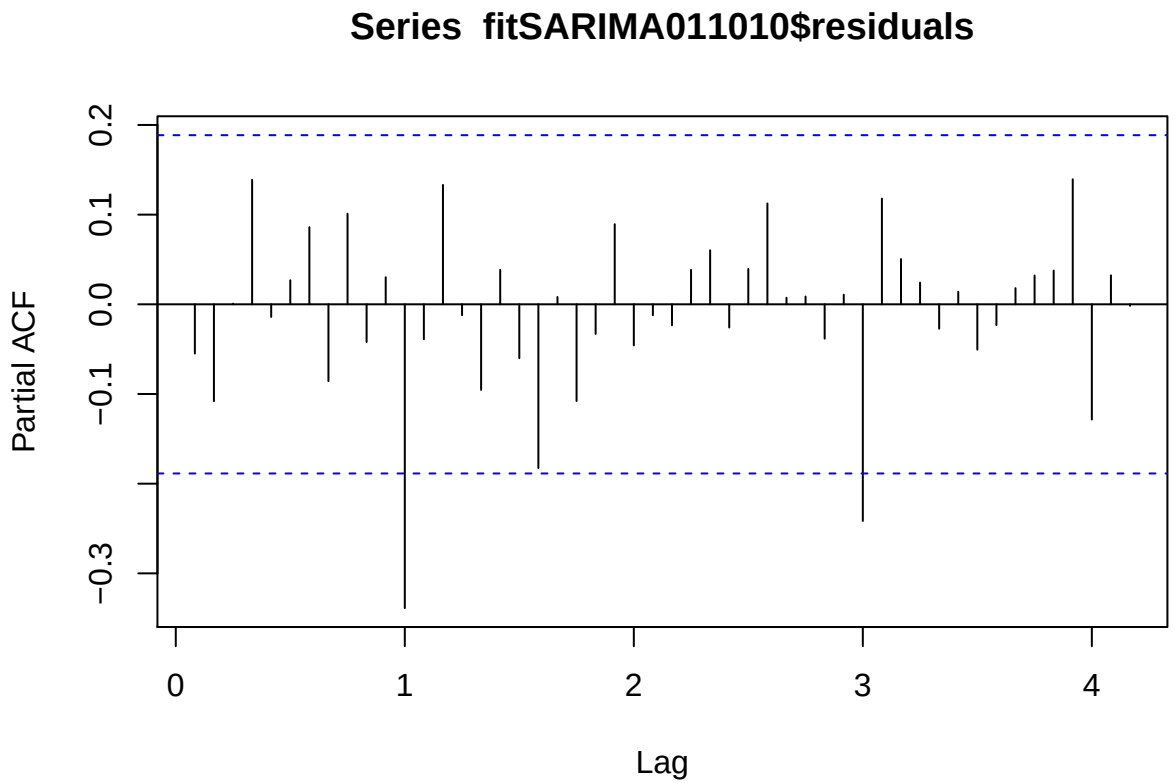
```
fitSARIMA011010 <- arima(Y.app,order=c(0,1,1),seasonal=list(order=c(0,1,0),period=12))
fitSARIMA011010
```

```
##
## Call:
## arima(x = Y.app, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 0), period = 12))
##
## Coefficients:
##           ma1
##       -0.8452
## s.e.   0.0598
##
## sigma^2 estimated as 0.01754:  log likelihood = 56.64,  aic = -109.28
```

```
acf(fitSARIMA011010$residuals,lag.max=50)
```




```
pacf(fitSARIMA011010$residuals,lag.max=50)
```



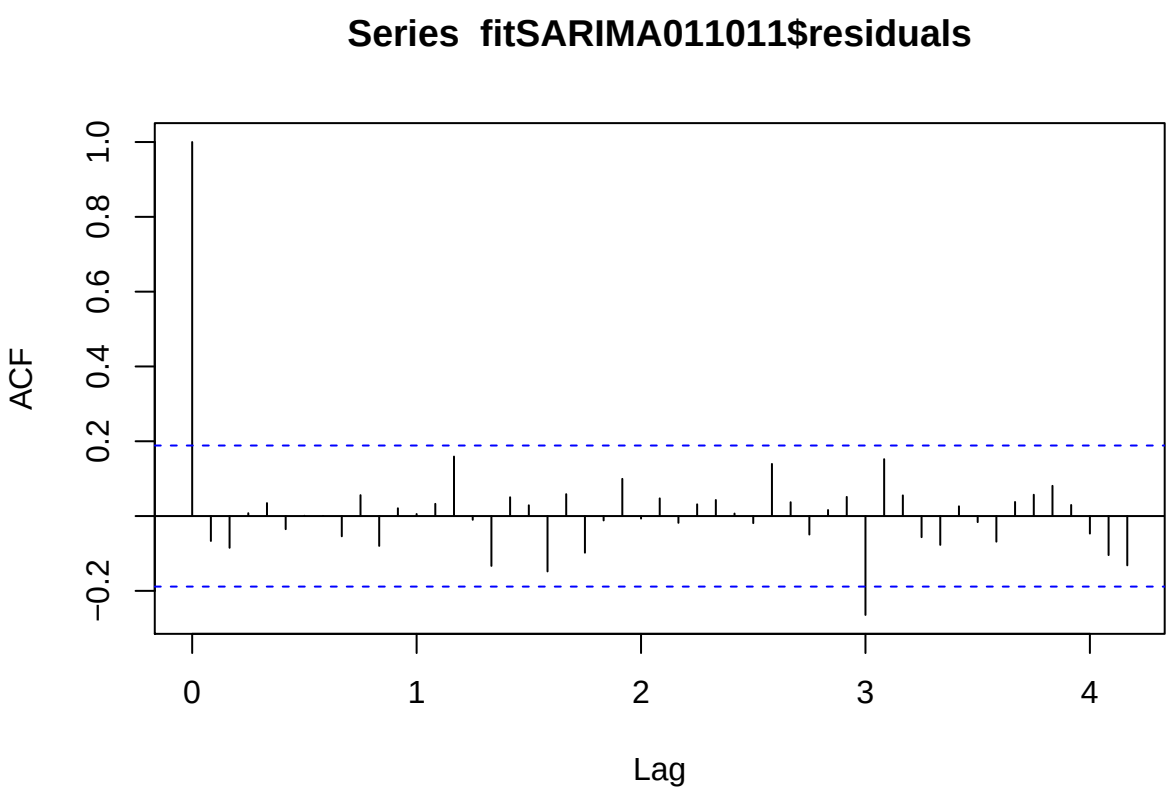
Les fonctions d'autocorrélation et d'autocorrélation partielle des résidus présentent des pics aux retards saisonniers, cela indique que la partie saisonnière de l'équation n'a pas été tout à fait correctement modélisée. Au vu de l'apparence de la fonction d'autocorrélation ressemblant à celle d'un processus MA, nous proposons

de poser $Q = 1$.

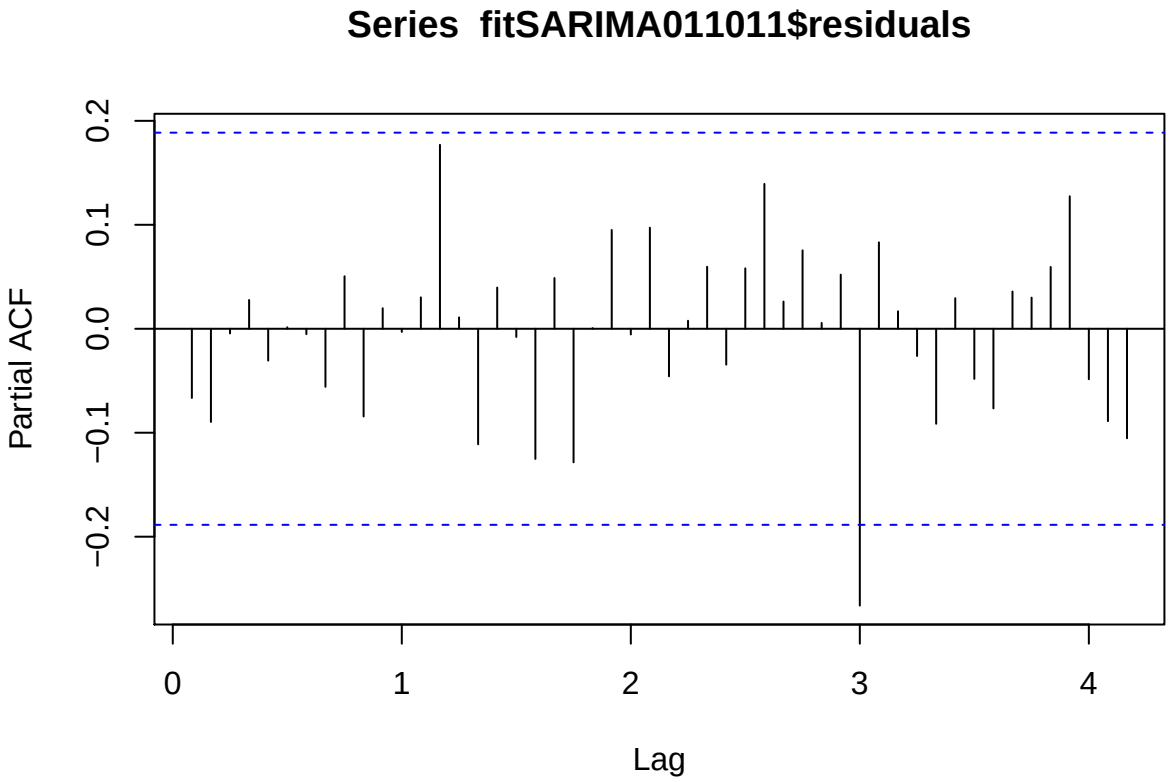
```
fitSARIMA011011 <- arima(Y.app,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=12))
fitSARIMA011011
```

```
##
## Call:
## arima(x = Y.app, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12))
##
## Coefficients:
##           ma1     sma1
##       -0.8058  -0.5998
## s.e.   0.0564   0.1380
##
## sigma^2 estimated as 0.01371:  log likelihood = 65.7,  aic = -125.4
```

```
acf(fitSARIMA011011$residuals,lag.max=50)
```




```
pacf(fitSARIMA011011$residuals,lag.max=50)
```

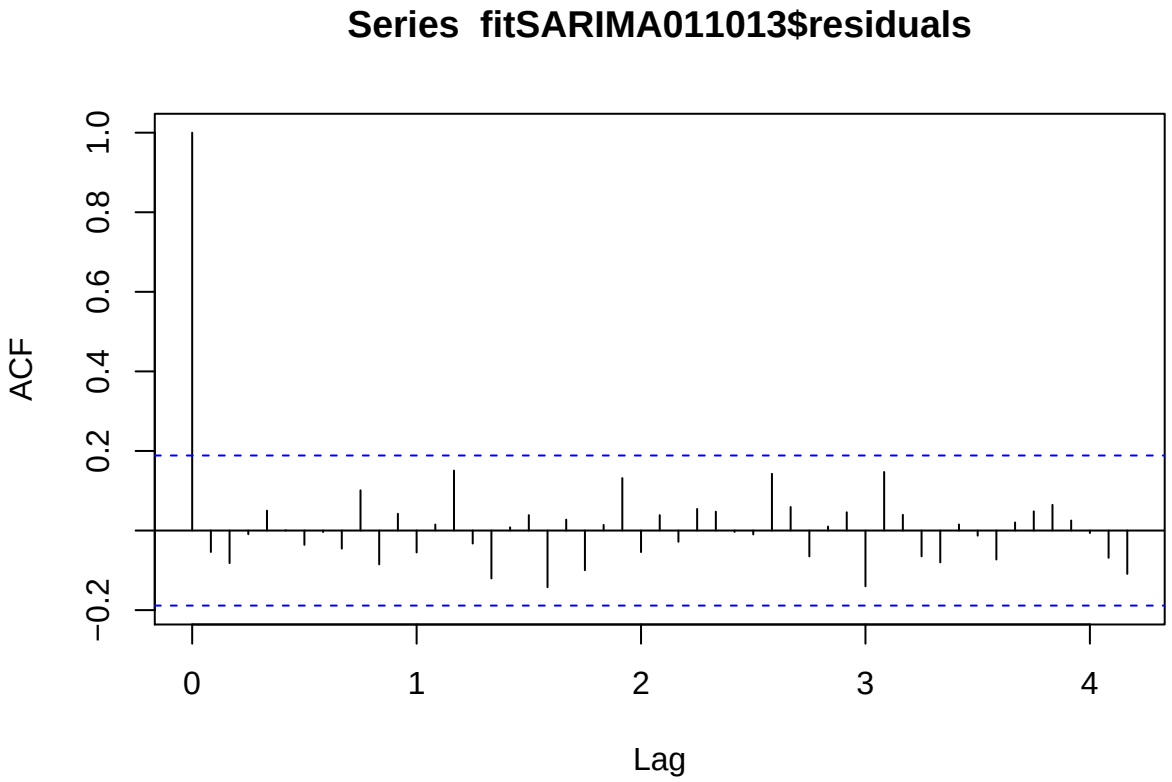


Etant la présence d'un pic restant au lag $3 * 12$ pour la fonction d'autocorrélation, nous posons finalement $Q = 3$, ce qui nous donne des résidus qui semblent être un bruit blanc.

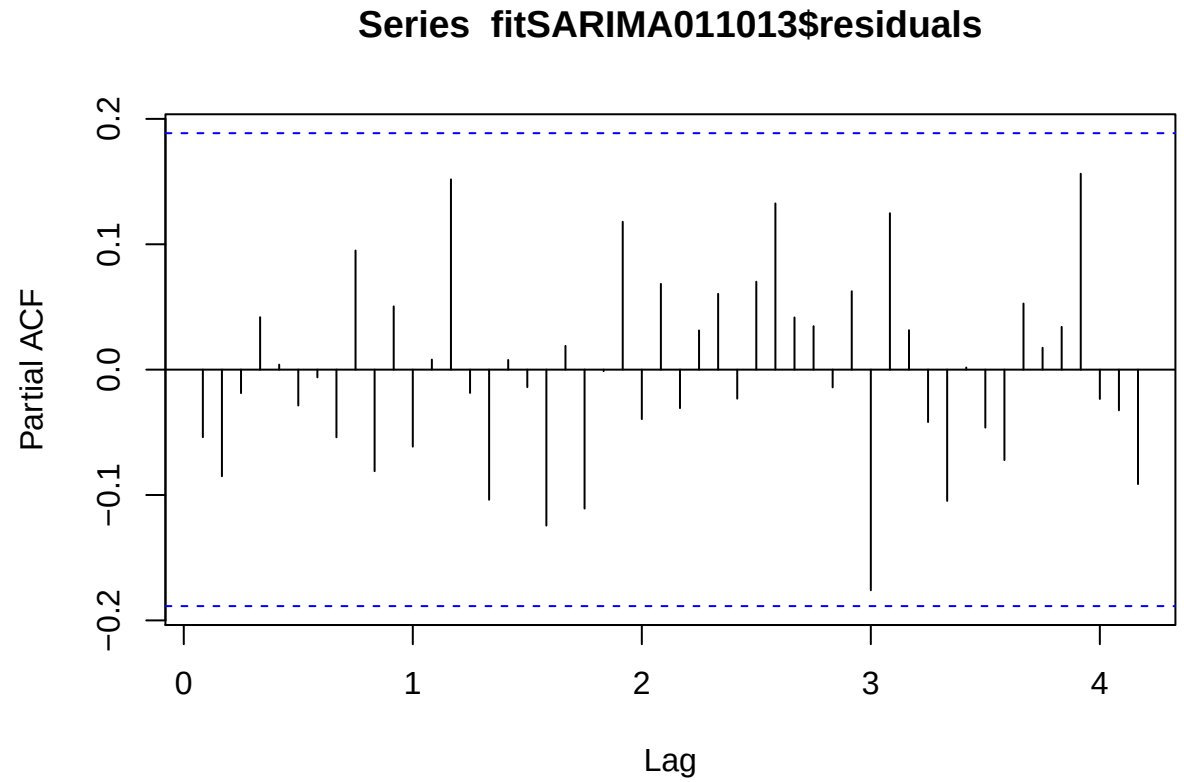
```
fitSARIMA011013 <- arima(Y.app,order=c(0,1,1),seasonal=list(order=c(0,1,3),period=12))
summary(fitSARIMA011013)
```

```
##
## Call:
## arima(x = Y.app, 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
## Training set -0.01441088 0.09942436 0.07049681 -0.1249002 0.5859839
##                   MASE        ACF1
## Training set 0.374361 -0.05387697
```

```
acf(fitSARIMA011013$residuals,lag.max=50)
```




```
pacf(fitSARIMA011013$residuals,lag.max=50)
```

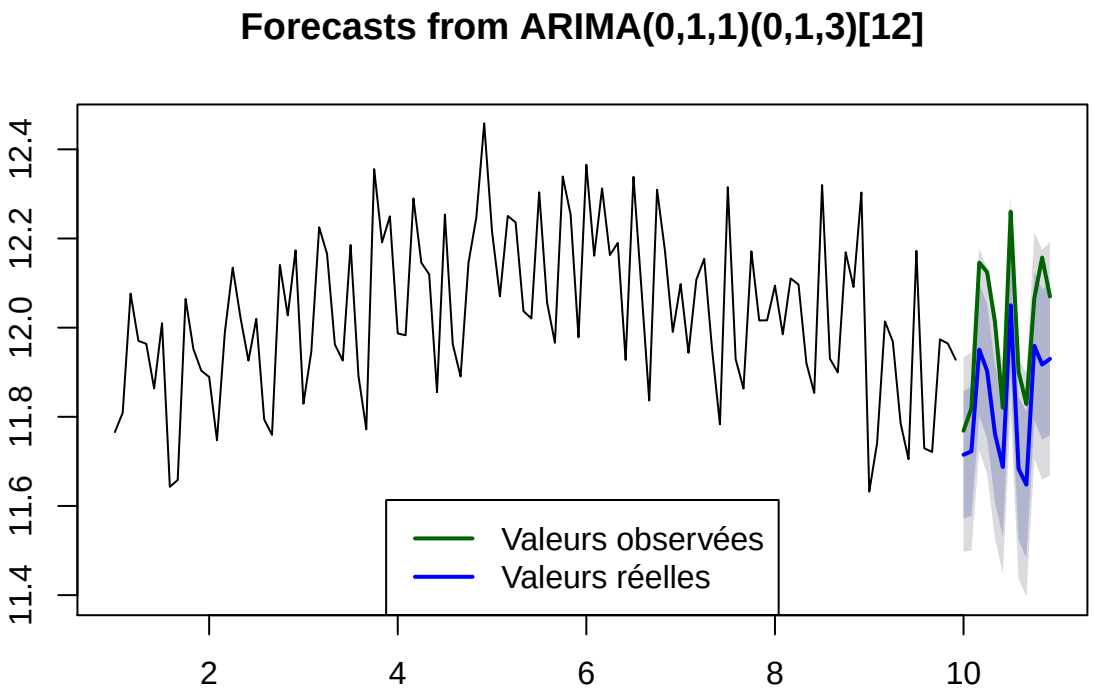


Nous retenons donc finalement un modèle

$$SARIMA(0,1,1)(0,1,3)_{12}$$

Observons maintenant les prédictions données par ce modèle :

```
predSARIMA011013 = forecast(fitSARIMA011013,h=12)
plot(predSARIMA011013)
points(Y.test,lwd=2,col="darkgreen",type='l')
legend('bottom',c('Valeurs observées','Valeurs réelles'),lty=rep(1,2),col=c('darkgreen','blue'),lwd=rep
```



Nous remarquons que les prédictions ont tendance à sous-estimer les valeurs réellement observées.

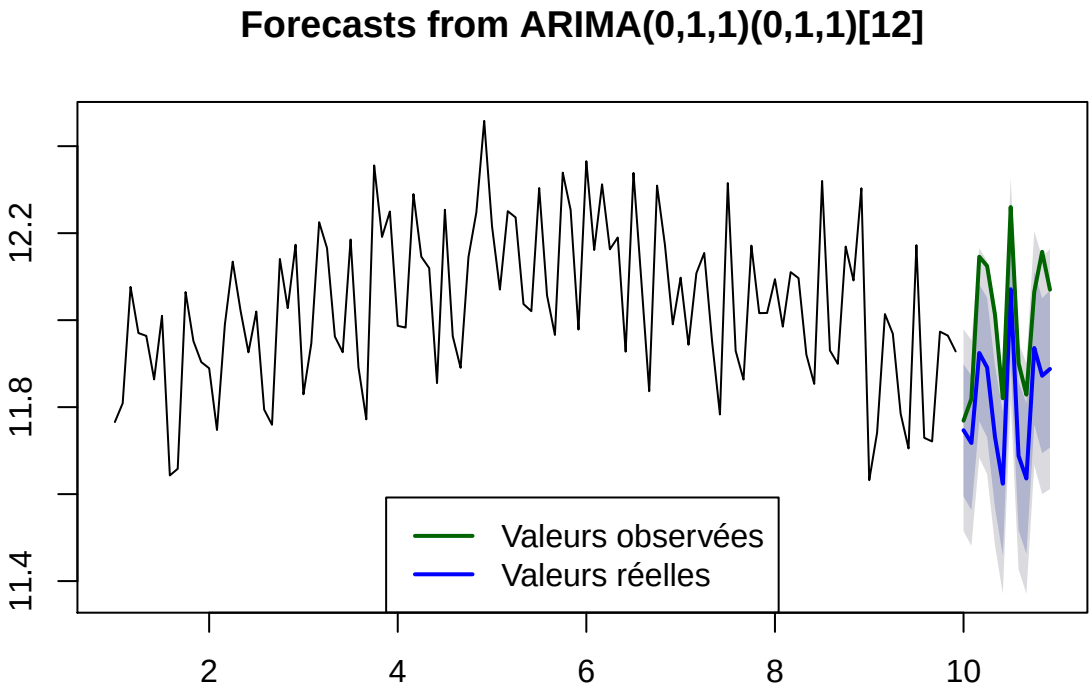
Nous comparons avec le modèle sélectionné automatiquement par la fonction `auto.arima` (remarquons par ailleurs que son AIC est plus élevé) :

```
fitSARIMA.IC <- auto.arima(Y.app)
summary(fitSARIMA.IC)
```

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

```
predSARIMA.IC = forecast(fitSARIMA.IC,h=12)
plot(predSARIMA.IC)
points(Y.test,lwd=2,col="darkgreen",type='l')
```

```
legend('bottom',c('Valeurs observées','Valeurs réelles'),lty=rep(1,2),col=c('darkgreen','blue'),lwd=rep
```



Le constat est le même, nous conservons pour le moment ces deux modèles et les comparerons aux prévisions obtenues par lissage exponentiel.

Lissage exponentiel

Nous comparons maintenant les résultats obtenus par estimation d'un processus de type SARIMA avec ceux obtenus par lissage exponentiel.

```
fit.ets <- ets(Y.app)
summary(fit.ets)
```

```
## ETS(M,N,A)
##
## Call:
##  ets(y = Y.app)
##
##   Smoothing parameters:
##     alpha = 0.2187
##     gamma = 1e-04
##
##   Initial states:
##     l = 11.976
##     s=0.0846 0.0673 0.1588 -0.1963 -0.1242 0.1901
##            -0.1548 -0.0458 0.0924 0.1041 -0.1095 -0.0666
##
##   sigma:  0.0085
##
##      AIC     AICc      BIC
## 42.44307 47.66046 82.67504
```

```
##
## Training set error measures:
##                       ME      RMSE        MAE         MPE      MAPE
## Training set -0.004090742 0.1021547 0.07596019 -0.04085735 0.6309863
##                    MASE         ACF1
## Training set 0.6631728 -0.004062284
```

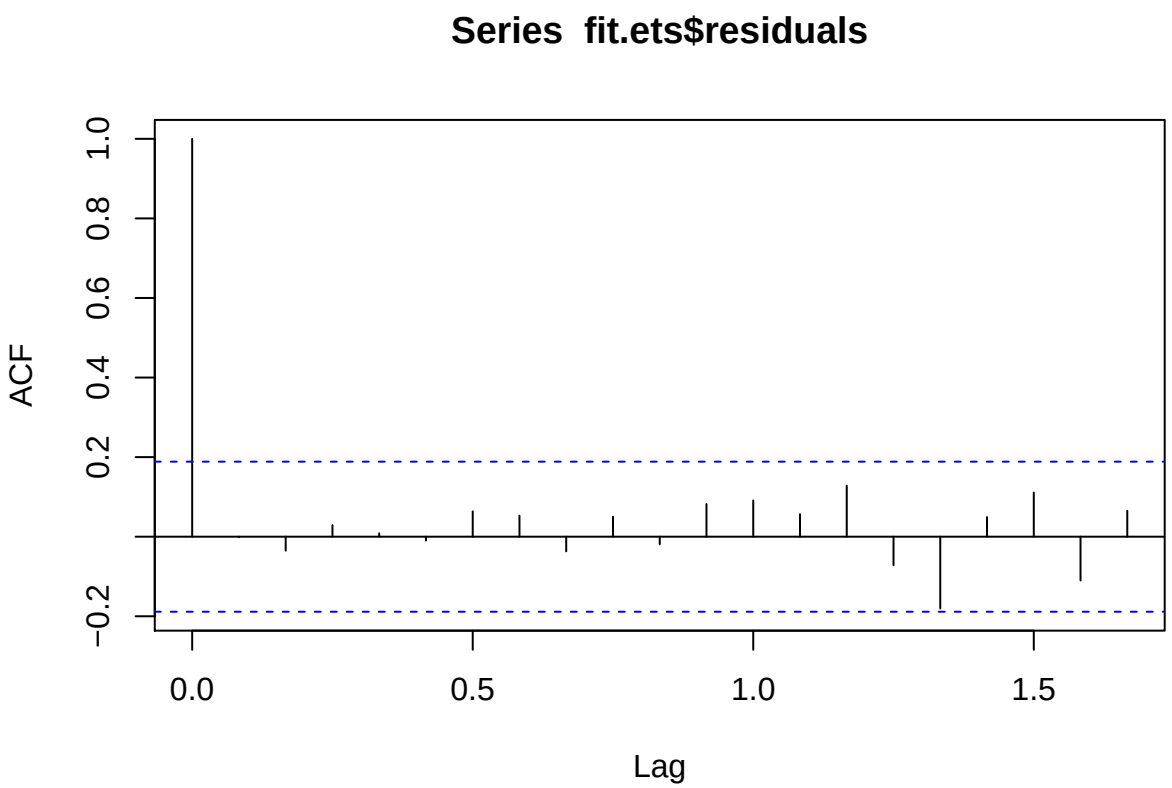
La fonction `ets()` sélectionne un modèle avec erreurs multiplicatives, sans tendance et avec saisonnalité additive, c'est-à-dire que les données sont supposées engendrées par le modèle espace-état suivant :

$$\begin{cases} X_t &= (\ell_{t-1} + s_{t-m})(1 + Z_t) \\ \ell_t &= \ell_{t-1} + \alpha(\ell_{t-1} + s_{t-m})Z_t \\ s_t &= s_{t-m} + \gamma(\ell_{t-1} + s_{t-m})Z_t, \end{cases}$$

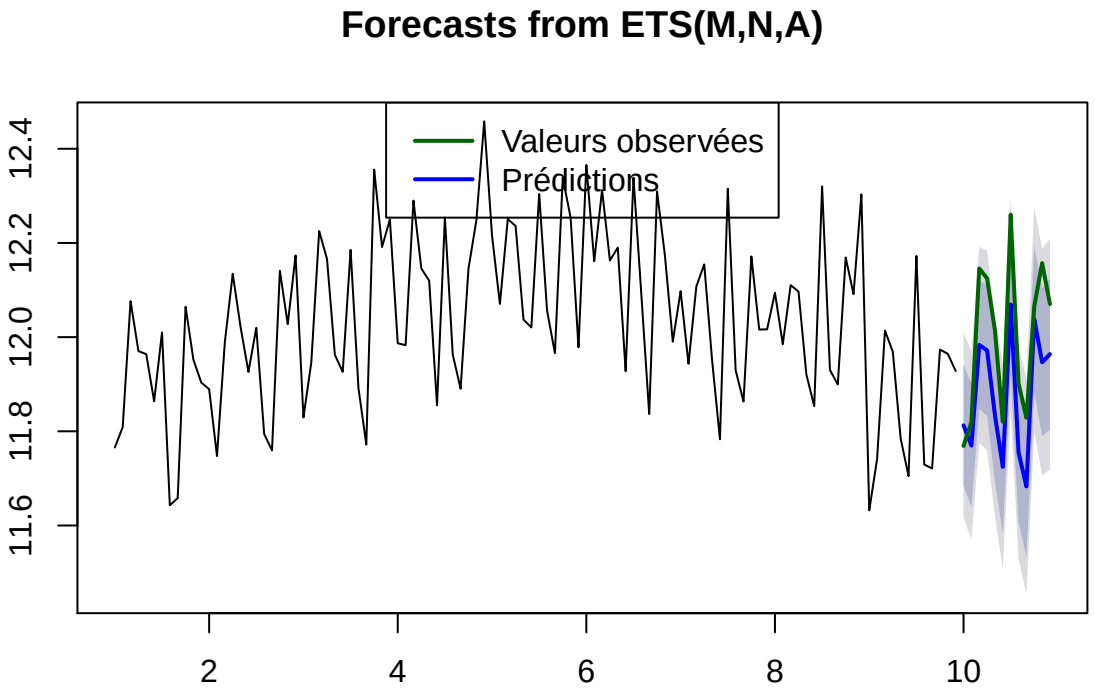
avec $(Z_t)_{t\in\mathbb{Z}}$ un bruit blanc de variance σ^2 et $m = 12$. Le résultat de la fonction `ets()` ci-dessous nous donne une estimation par maximum de vraisemblance des paramètres α, γ et σ ainsi que des valeurs initiales ℓ_0, $(s_{-11}, ..., s_0)$.

Que doit-on modifier dans l'équation précédente pour obtenir un modèle espace-état avec erreurs additives, sans tendance et avec saisonnalité additive ?

```
acf(fit.ets$residuals)
```




```
pred.ets <- forecast(fit.ets,h=12)
plot(pred.ets)
points(Y.test,type='l',col='darkgreen',lwd=2)
legend('top',c("Valeurs observées","Prédictions"),col=c("darkgreen","blue"),lty=rep(1,2),lwd = rep(2,2))
```

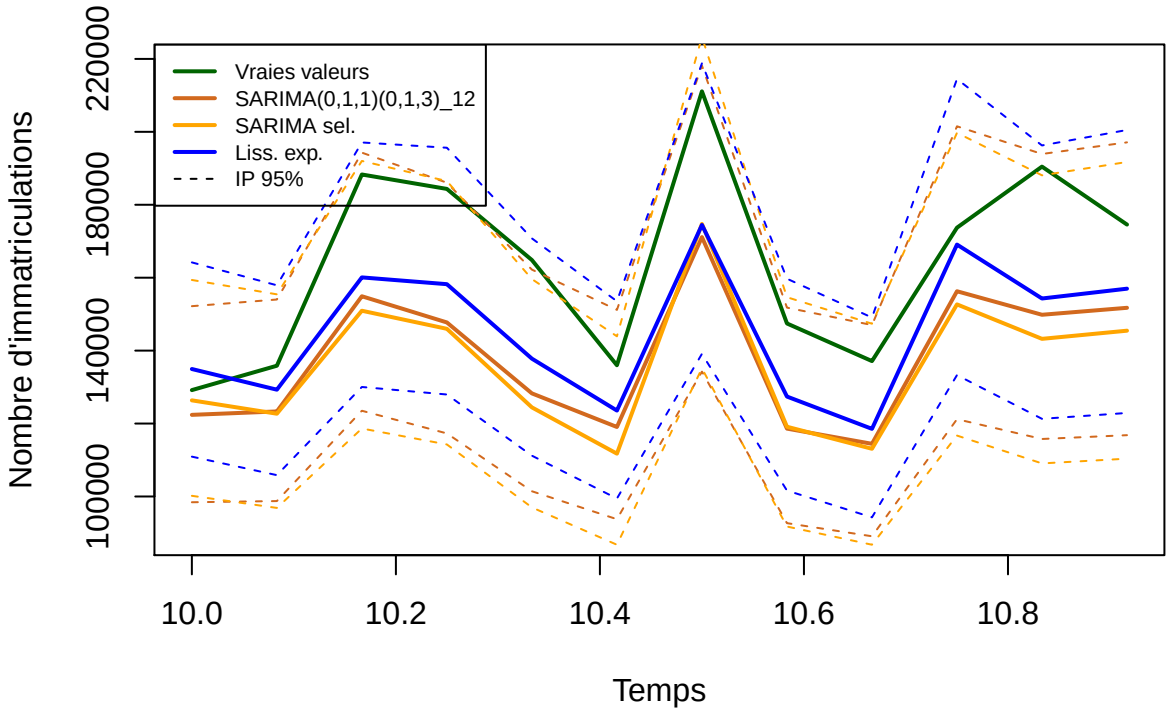


Encore une fois il semble que l'on sous-estime un peu les prédictions mais la méthode de lissage expoentiel basée sur le modèle ci-dessus semble donner de meilleures prédictions que celle basée sur l'estimation d'un modèle SARIMA.

Comparaison des deux approches :

Nous traçons tout d'abord les prédictions et les valeurs réelles sur la série initiale $X_t = \exp(Y_t)$.

```
plot(X.test, col = "darkgreen", lwd = 2, ylab = "Nombre d'immatriculations", xlab = "Temps",
    ylim = range(c(X.test, exp(predSARIMA011013$lower), exp(predSARIMA011013$upper),
        exp(pred.ets$lower), exp(pred.ets$upper))))
points(exp(predSARIMA011013$mean), col = "chocolate", lwd = 2, type = "l")
points(exp(predSARIMA011013$lower[, 2]), col = "chocolate", type = "l", lty = 2)
points(exp(predSARIMA011013$upper[, 2]), col = "chocolate", type = "l", lty = 2)
points(exp(predSARIMA.IC$mean), col = "orange", lwd = 2, type = "l")
points(exp(predSARIMA.IC$lower[, 2]), col = "orange", type = "l", lty = 2)
points(exp(predSARIMA.IC$upper[, 2]), col = "orange", type = "l", lty = 2)
points(exp(pred.ets$mean), col = "blue", lwd = 2, type = "l")
points(exp(pred.ets$lower[, 2]), col = "blue", type = "l", lty = 2)
points(exp(pred.ets$upper[, 2]), col = "blue", type = "l", lty = 2)
legend("topleft", c("Vraies valeurs", "SARIMA(0,1,1)(0,1,3)_12", "SARIMA sel.", "Liss. exp.",
    "IP 95%"), col = c("darkgreen", "chocolate", "orange", "blue", "black"), lty = c(rep(1,
    4), 2), lwd = c(rep(2, 4), 1), cex = 0.7)
```



Quel modèle propseriez-vous pour la série ? Écrire la série de commandes R permettant de prédire le nombre d'immatriculations de la 11ème année (non observée). Ne pas oublier de réestimer les paramètres du modèle choisi.

Nous constatons que le lissage exponentiel donne, pour cette série-là, de meilleures prévisions. Remarquons également que le modèle $SARIMA(0,1,1)(0,1,3)_{12}$ que nous avons déduit de l'observation des fonctions d'autocorrélations donne de meilleure prévisions que le modèle sélectionné automatiquement.