Rapport I

Olivier Berthier

2024-05-31

Ce projet présente l'analyse d'une série temporelle concernant le nombre de passagers sur le réseau SNCF. Parmi les séries proposées, la forme particulière de son évolution dans le temps a retenu notre attention.

```
library(zoo)
library(tseries)
library(ggseas)
library(caschrono)
library(forecast)
library(ggplot2)
library(lmtest)
library(car)
library (MASS)
```

Importation et exploration des données

```
sncf=read.table("http://freakonometrics.free.fr/sncf.csv",header=TRUE,sep=";")
head (sncf)
```

```
##   ANNEE JANVIER FEVRIER MARS AVRIL  MAI JUIN JUILLET AOUT SEPTEMBRE OCTOBRE
## 1  1963    1750    1560 1820  2090 1910 2410    3140 2850      2090    1850
## 2  1964    1710    1600 1800  2120 2100 2460    3200 2960      2190    1870
## 3  1965    1670    1640 1770  2190 2020 2610    3190 2860      2140    1870
## 4  1966    1810    1640 1860  1990 2110 2500    3030 2900      2160    1940
## 5  1967    1850    1590 1880  2210 2110 2480    2880 2670      2100    1920
## 6  1968    1834    1792 1860  2138 2115 2485    2581 2639      2038    1936
##   NOVEMBRE DECEMBRE
## 1     1630     2420
## 2     1770     2270
## 3     1760     2360
## 4     1750     2330
## 5     1670     2520
## 6     1784     2391
```

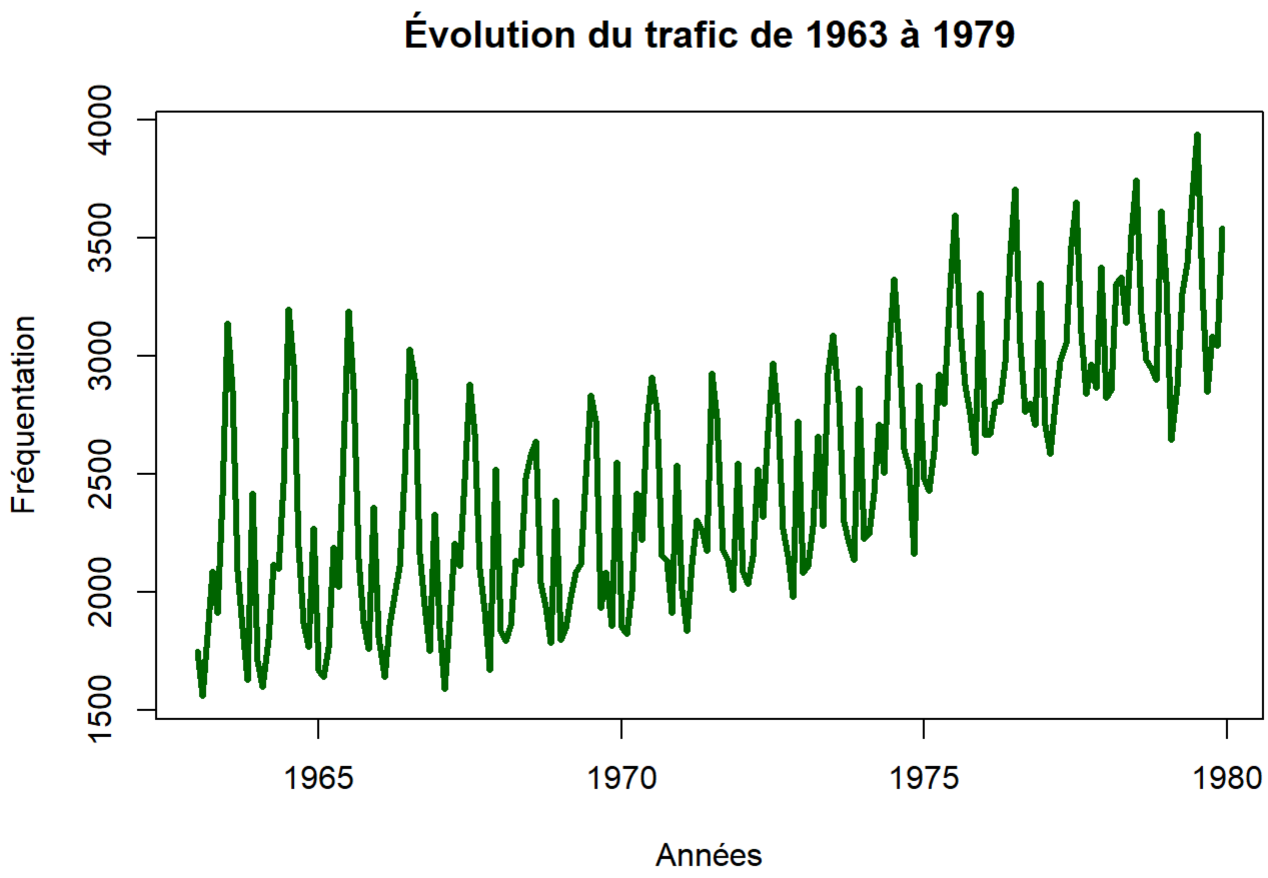
```
summary(sncf)
```

```
##      ANNEE        JANVIER       FEVRIER          MARS          AVRIL
##  Min.   :1963   Min.   :1670   Min.   :1560   Min.   :1770   Min.   :1990
##  1st Qu.:1967   1st Qu.:1816   1st Qu.:1678   1st Qu.:1865   1st Qu.:2151
##  Median :1972   Median :2044   Median :1942   Median :2136   Median :2470
##  Mean   :1972   Mean   :2195   Mean   :2101   Mean   :2309   Mean   :2555
##  3rd Qu.:1976   3rd Qu.:2620   3rd Qu.:2546   3rd Qu.:2746   3rd Qu.:2894
##  Max.   :1980   Max.   :3313   Max.   :2913   Max.   :3306   Max.   :3333
##       MAI           JUIN         JUILLET          AOUT        SEPTEMBRE
##  Min.   :1910   Min.   :2175   Min.   :2581   Min.   :2639   Min.   :1932
##  1st Qu.:2111   1st Qu.:2486   1st Qu.:2939   1st Qu.:2762   1st Qu.:2143
##  Median :2272   Median :2703   Median :3165   Median :2880   Median :2228
##  Mean   :2489   Mean   :2888   Mean   :3249   Mean   :2928   Mean   :2426
##  3rd Qu.:2931   3rd Qu.:3394   3rd Qu.:3636   3rd Qu.:3084   3rd Qu.:2820
##  Max.   :3391   Max.   :3682   Max.   :3937   Max.   :3284   Max.   :3206
##     OCTOBRE        NOVEMBRE       DECEMBRE
##  Min.   :1850   Min.   :1630   Min.   :2270
##  1st Qu.:1937   1st Qu.:1774   1st Qu.:2445
##  Median :2144   Median :1994   Median :2638
##  Mean   :2359   Mean   :2205   Mean   :2861
##  3rd Qu.:2790   3rd Qu.:2677   3rd Qu.:3297
##  Max.   :3269   Max.   :3181   Max.   :4008
```

Les données concernent la fréquentation mensuelle au sein de la SNCF étalée sur 18 ans, de 1963 à 1980. Nous ne connaissons pas l'unité utilisée.
Le *summary* nous indique déjà une forte saisonnalité. La moyenne la plus forte se situe au mois de juillet, la plus faible en février (aussi le mois le plus court).

Transformation du dataframe en données de type série temporelle (ts) et création de deux jeux, l'un d'entraînement (**d_train**), l'autre de test (**d_test**). Nous isolons l'année 1980.

```
train=as.vector(t(as.matrix(sncf[,2:13])))
d <- ts(train,start = c(1963, 1), frequency = 12)
# Division en deux jeux
d_train <- window(d, end = c(1979, 12))   # Jeu d'entraînement de janvier 1963 à décembre 197
9
d_test <- window(d, start = c(1980, 1))   # Jeu de test de janvier 1980 à décembre 1980
plot (d_train, col ="darkgreen", ylab="Fréquentation", xlab = "Années", lwd= 3)  # Graphique
du jeu d'entraînement
title(main = "Évolution du trafic de 1963 à 1979")
```



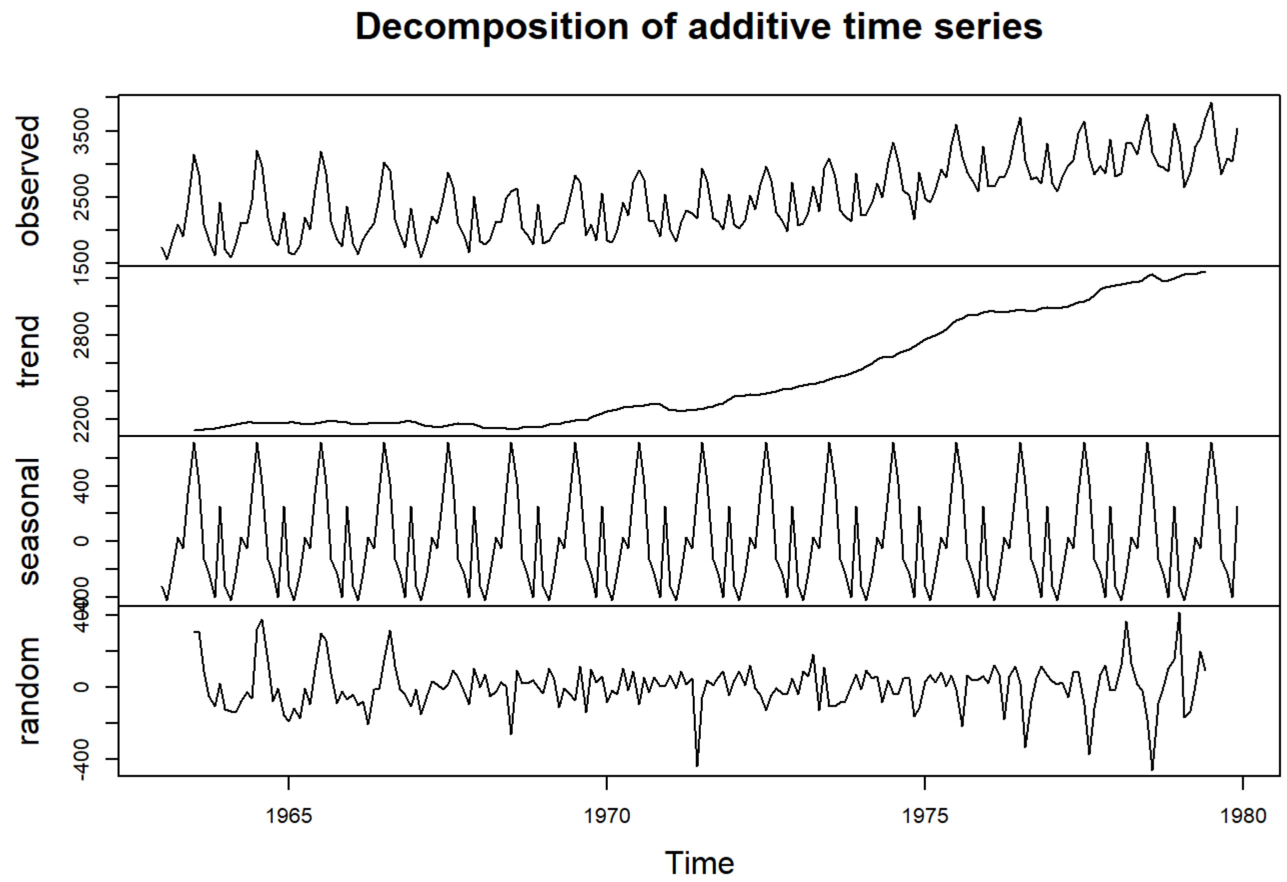
On remarque une stabilité de la moyenne jusqu'en 1970, puis une tendance nette à la hausse jusqu'en 1980. Notons que cette tendance à la hausse est moins marquée les 5 dernières années. Une forte saisonnalité court tout au long de la série.

Sur l'ensemble de la période, les données ont un comportement globalement **hétéroscédastique**, i.e.: la variance la plus forte s'observe les premières années, elle se réduit ensuite pour atteindre ce qui semble être un minimum en 1968, puis augmente à nouveau en 1969 pour adopter un comportement relativement **homoscédastique**.

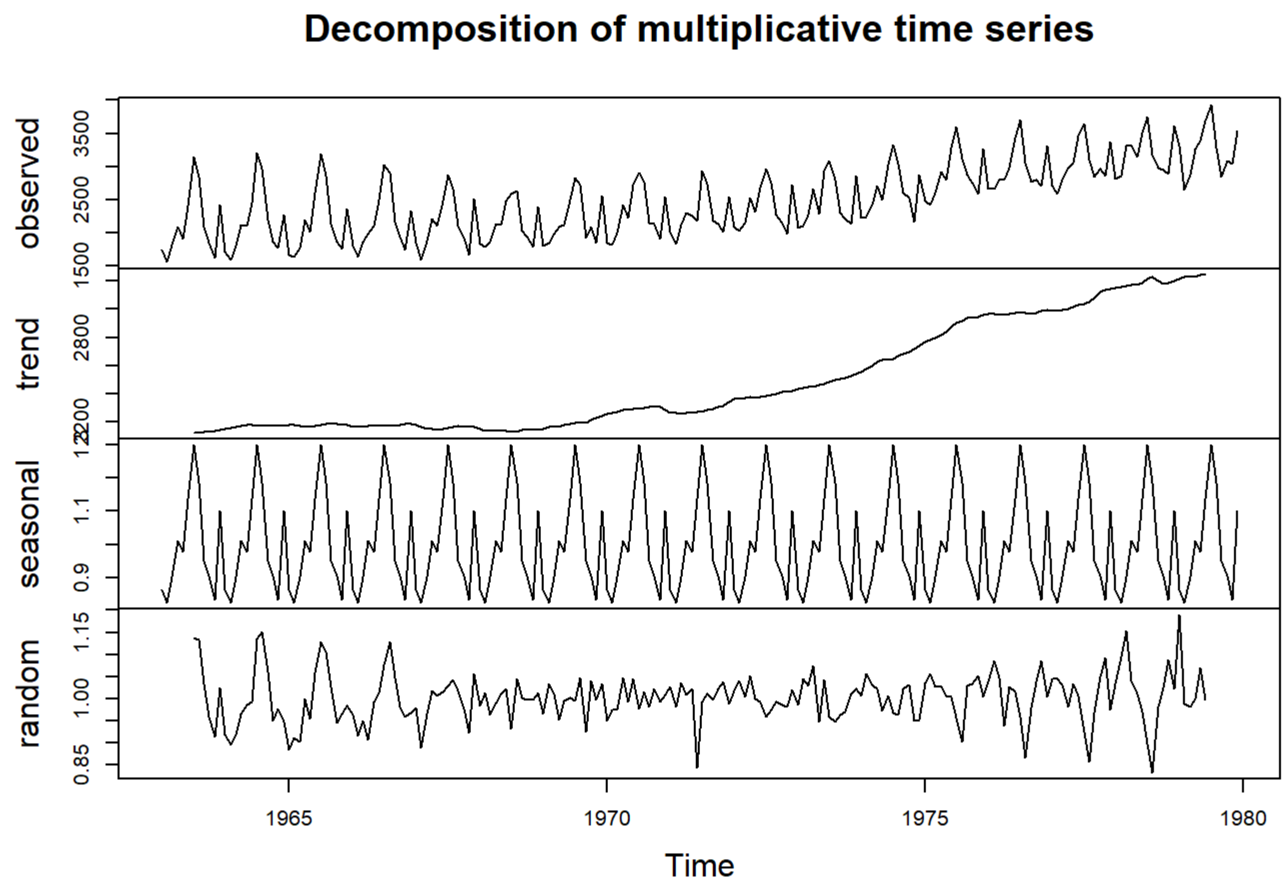
Nous avons essayé plusieurs transformations de la variable y (racine carrée, log, quadratique, différentes valeurs de γ dans le cadre de transformations Box-Cox, etc). Aucune ne donne de résultats probants.

Décomposons la série:

```
decomp_ad <- decompose(d_train, "additive")
decomp_mul <- decompose(d_train, "multiplicative")
plot (decomp_ad)
```

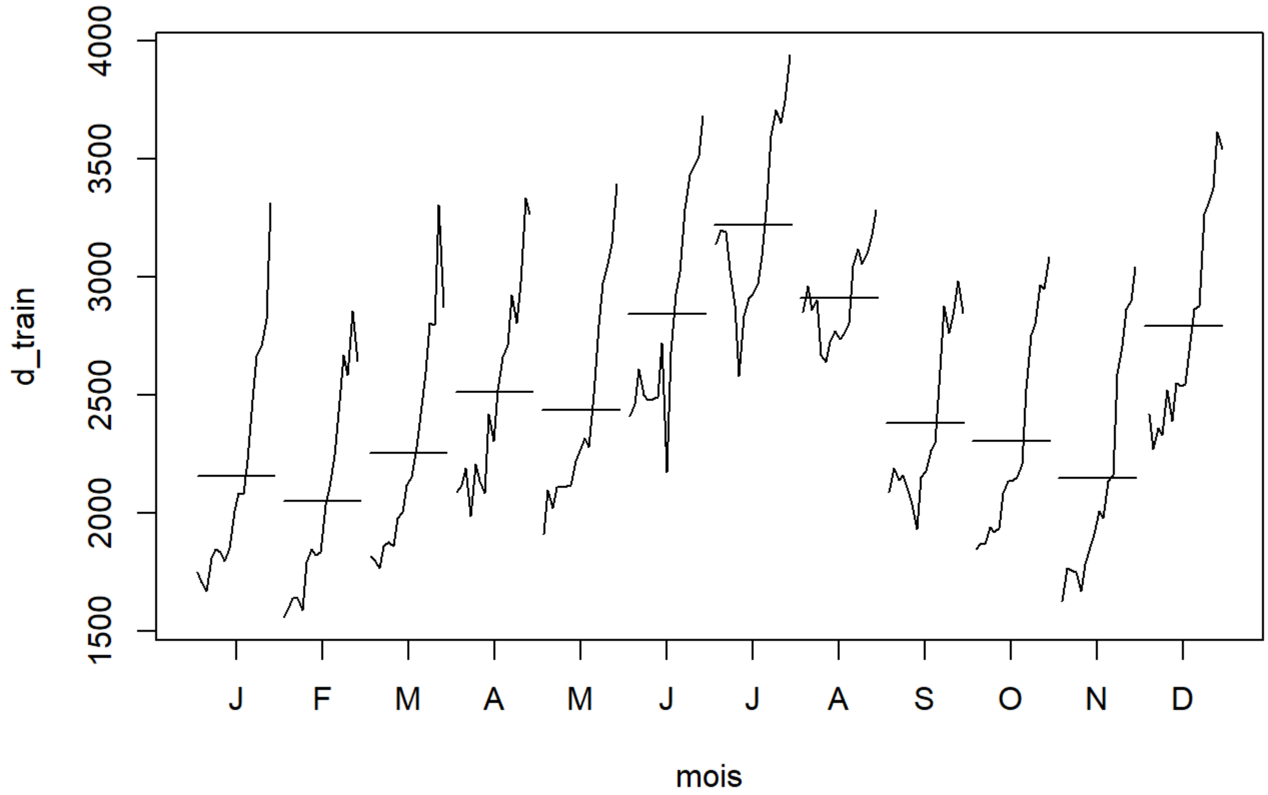



```
plot (decomp_mul)
```



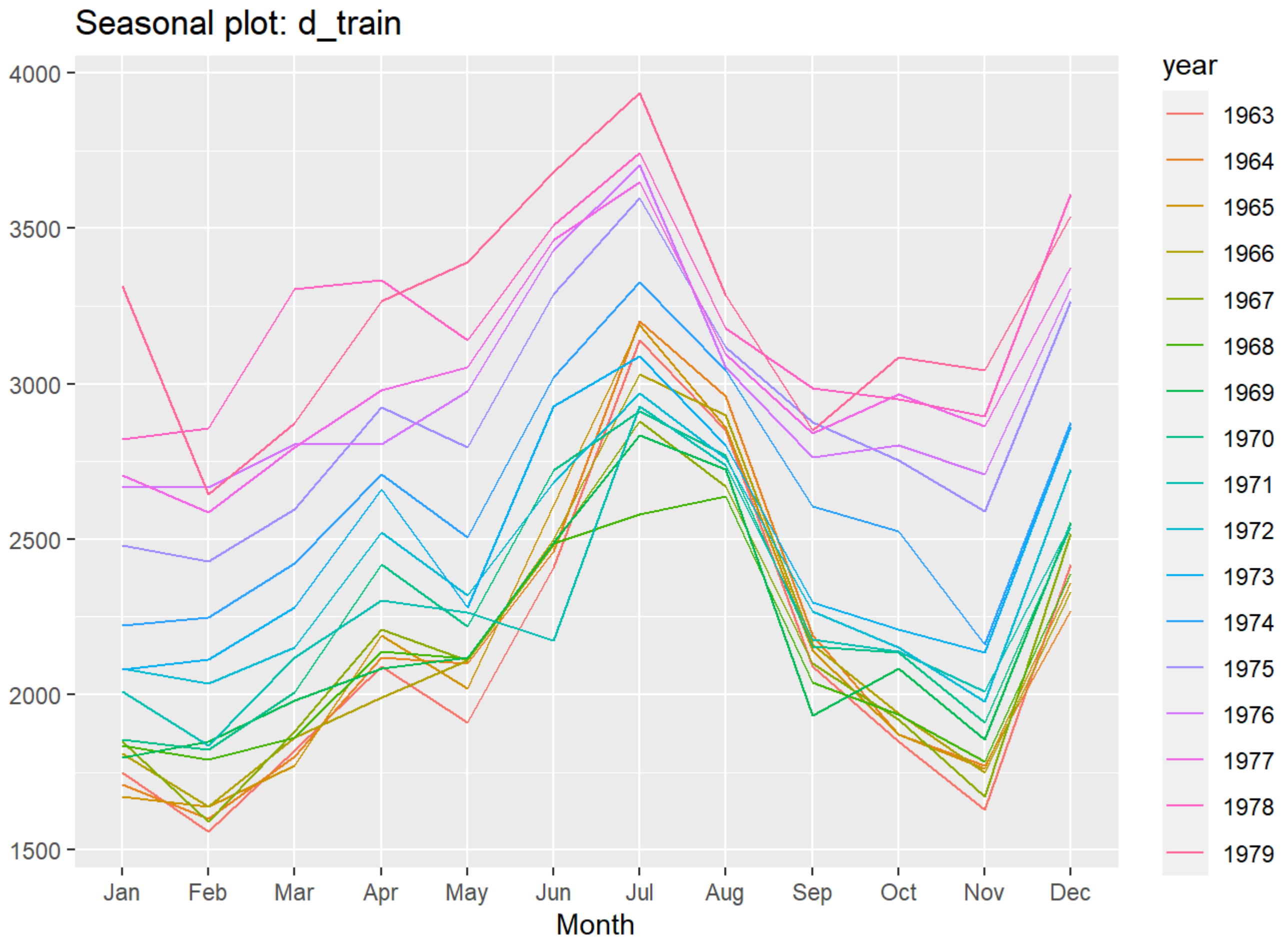
La fonction *decompose()* confirme que le modèle n’est pas purement multiplicatif.

```
monthplot(d_train,xlab='mois')
```



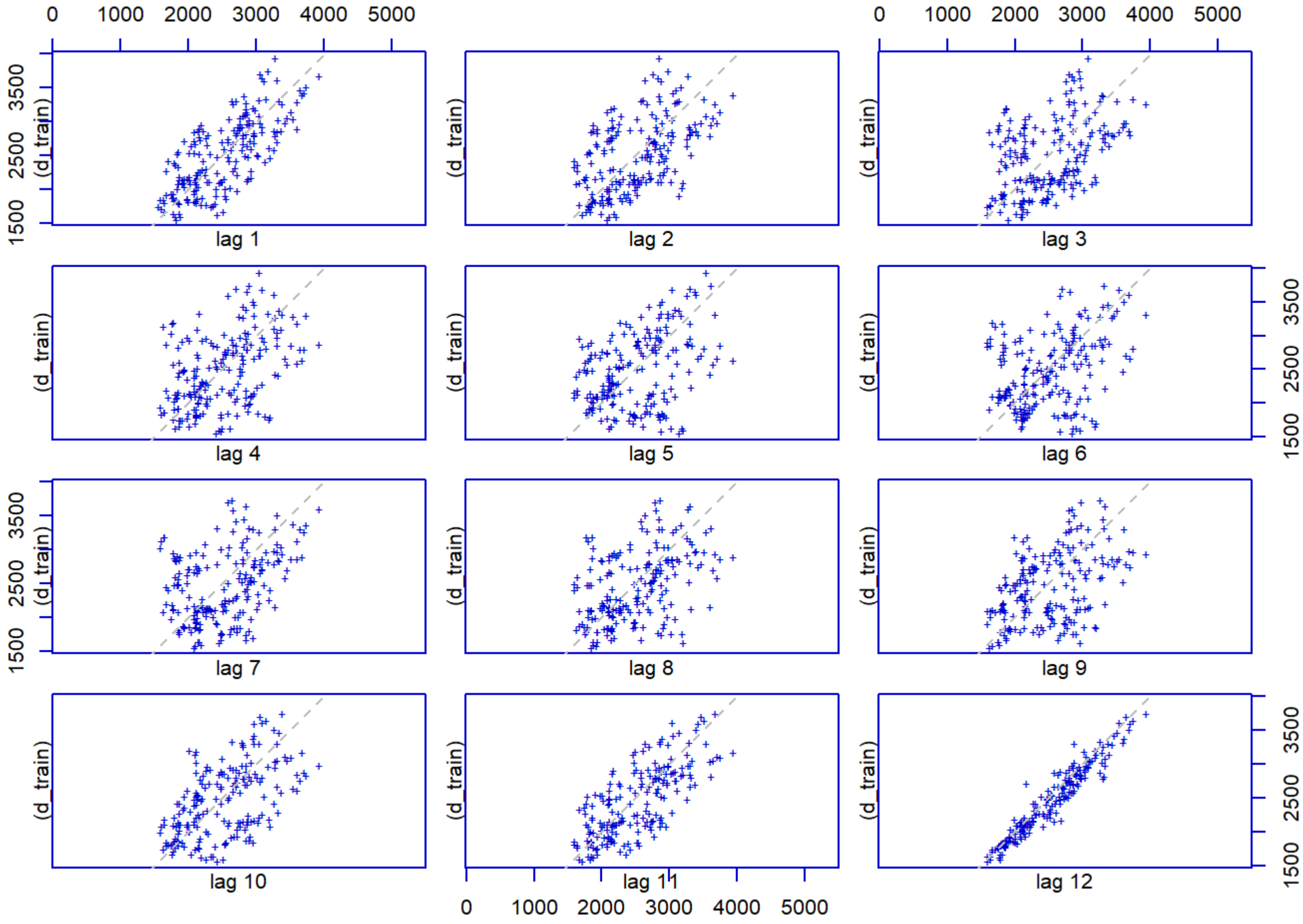
Le *monthplot* de la série confirme la présence d’une importante saisonnalité.
Sans surprise, la fréquentation est maximale pendant la saison estivale (juin, juillet, août), un pic isolé est aussi présent en décembre. Les vacances d’été et les fêtes de fin d’année peuvent expliquer ces chiffres.

```
# Création du "seasonality plot"
ggseasonplot(d_train, date_col = "Date", value_col = "Value", freq = 12) # freq=12 pour une f
réquence mensuelle
```



Le *seasonality plot* confirme la présence des deux pics annuels de fréquentation. Ce graphique met en valeur les “creux” de fréquentation de février et novembre (surtout les premières années).
On distingue difficilement le type de modèle: additif ou multiplicatif. En effet, on peut observer une tendance multiplicative les premières années (les pics s’accentuent), puis une période de type plutôt “additif” à partir du début des années 70 (les courbes bleues et au-dessus).

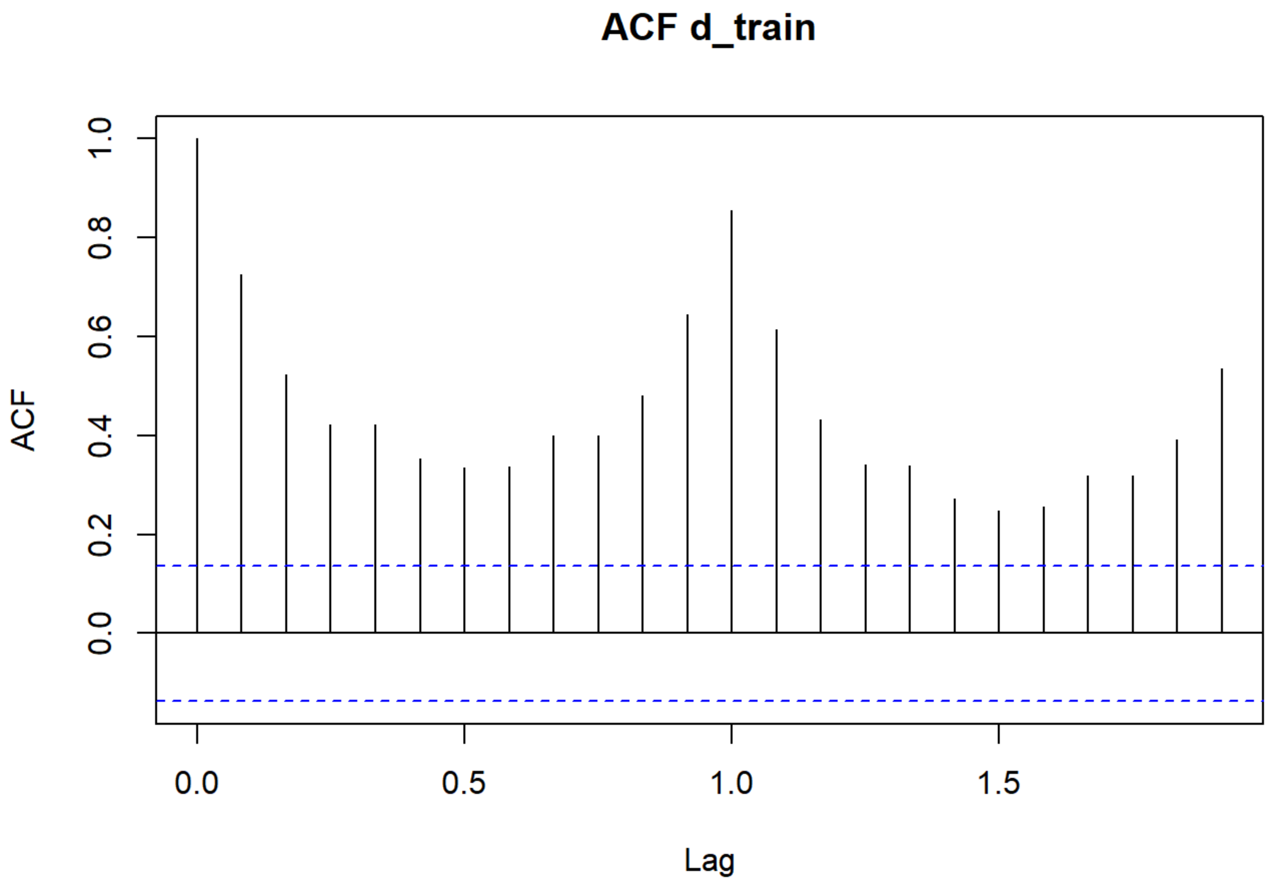
```
lag.plot((d_train),set=c(1:12),do.lines=FALSE, pch="+",col="blue3") # Affichage des "lagplot" de 1 à 12:
```



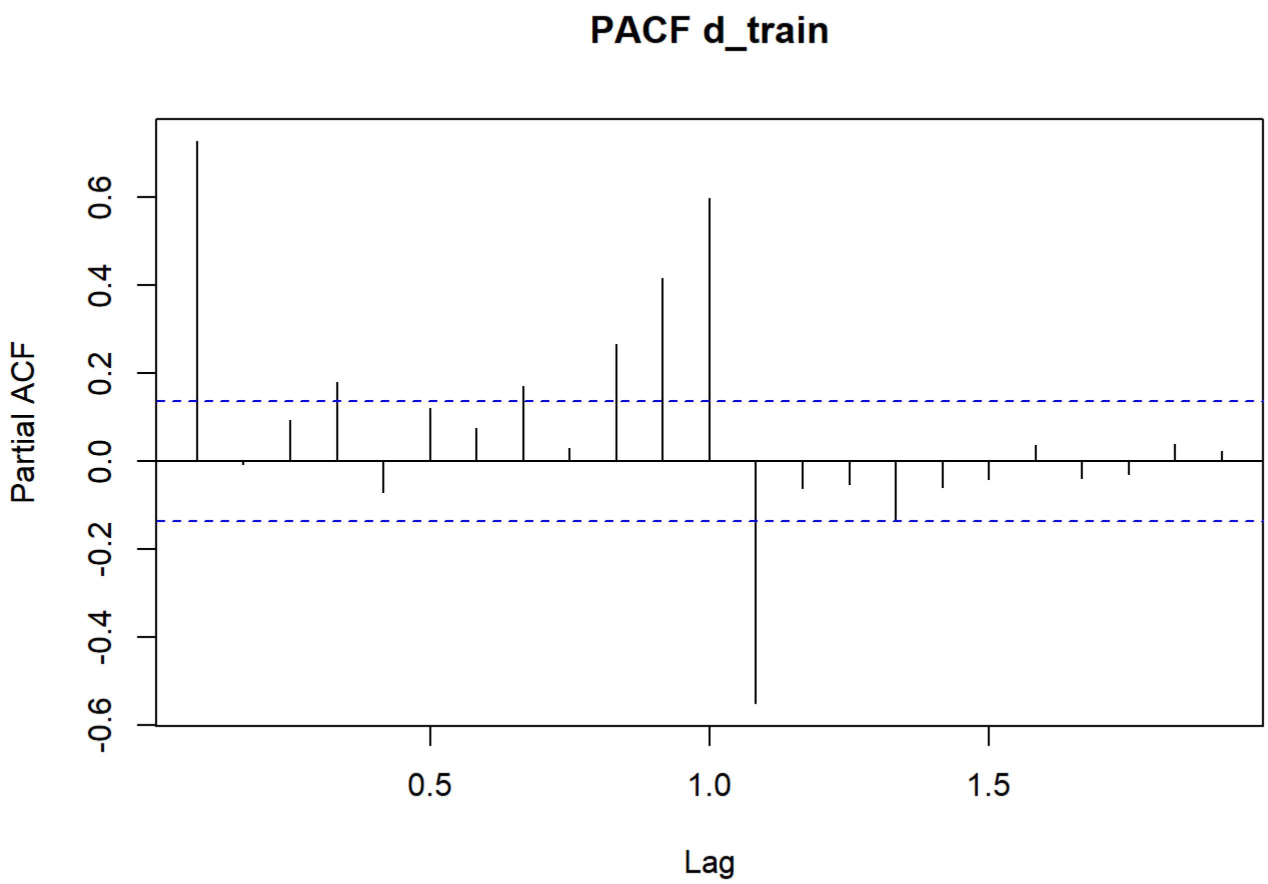
Corrélation d’ordre 12 évidente (sans surprise).

Autocorrélation

```
# Affichage des graphiques d'autocorrélation et d'autocorrélation partielle
acf(d_train, main = "ACF d_train")
```




```
pacf (d_train, main = "PACF d_train")
```

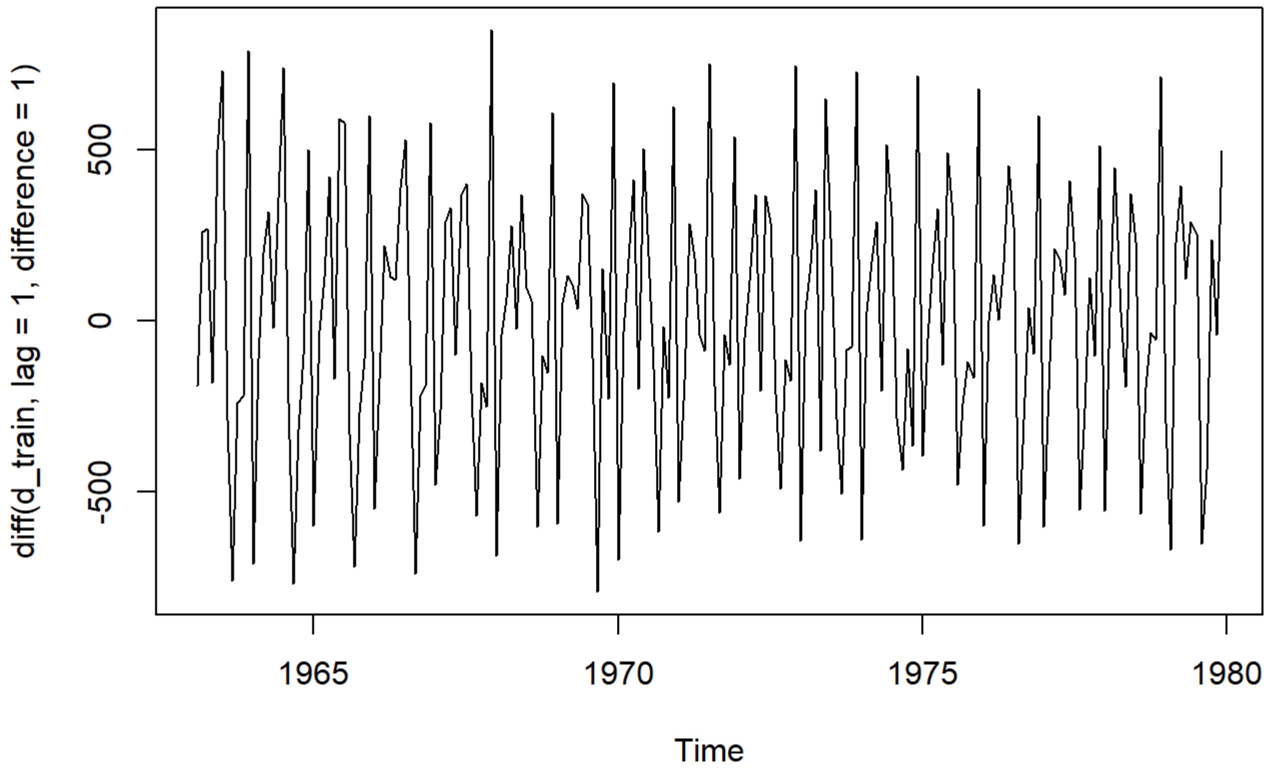


Les graphiques sont typiques d'une série à forte saisonnalité.
La série nécessite une stationnarisation avant de la modéliser.

Stationnarisation de la série

Commençons par éliminer la tendance par différenciation d'ordre 1:

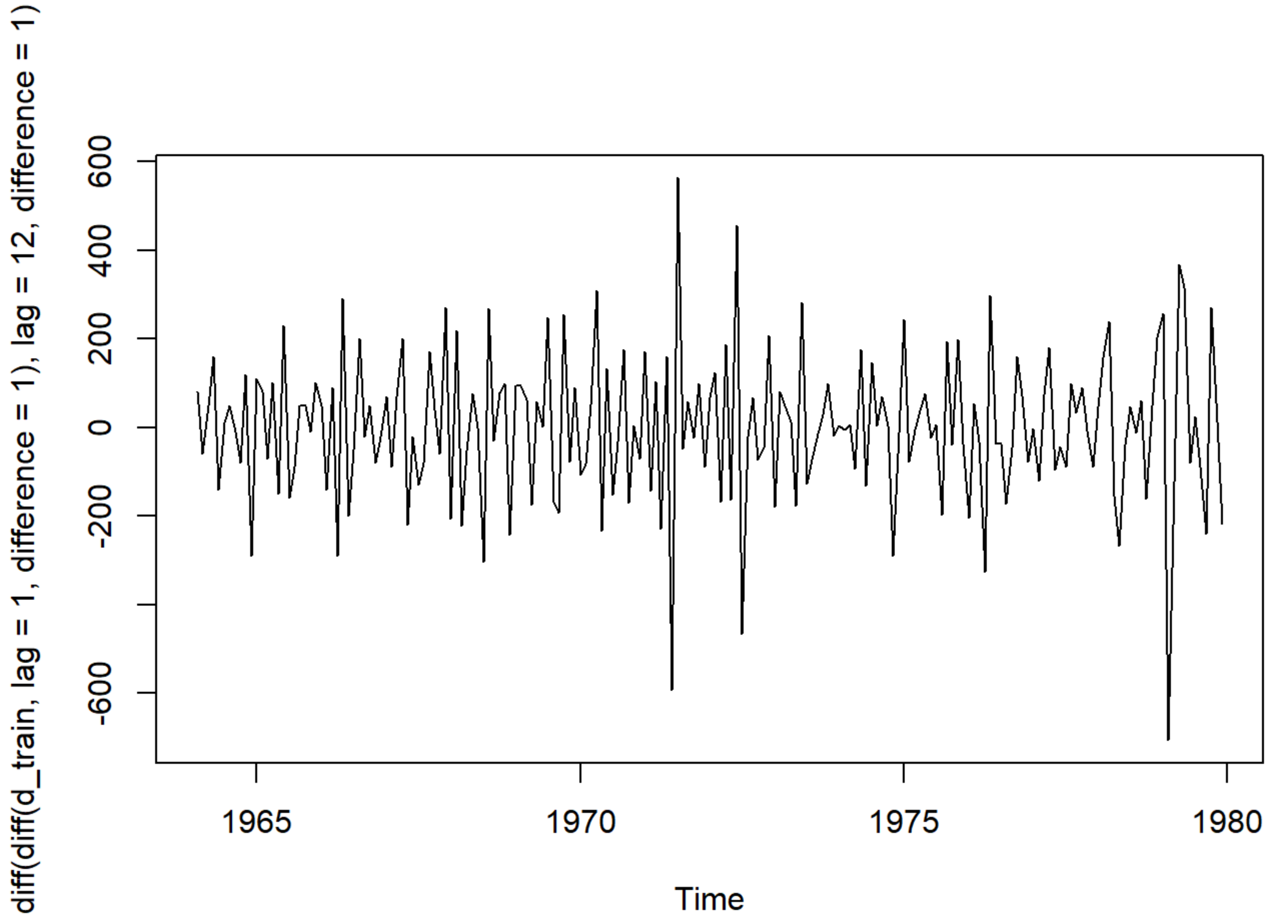
```
plot(diff(d_train,lag=1,difference=1))
```



La tendance à la hausse (le trend) est bel et bien éliminée. (Nous avons essayé d'autres ordres de différenciation, mais le meilleur résultat est obtenu avec l'ordre 1.)

Élimination de la saisonnalité:

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

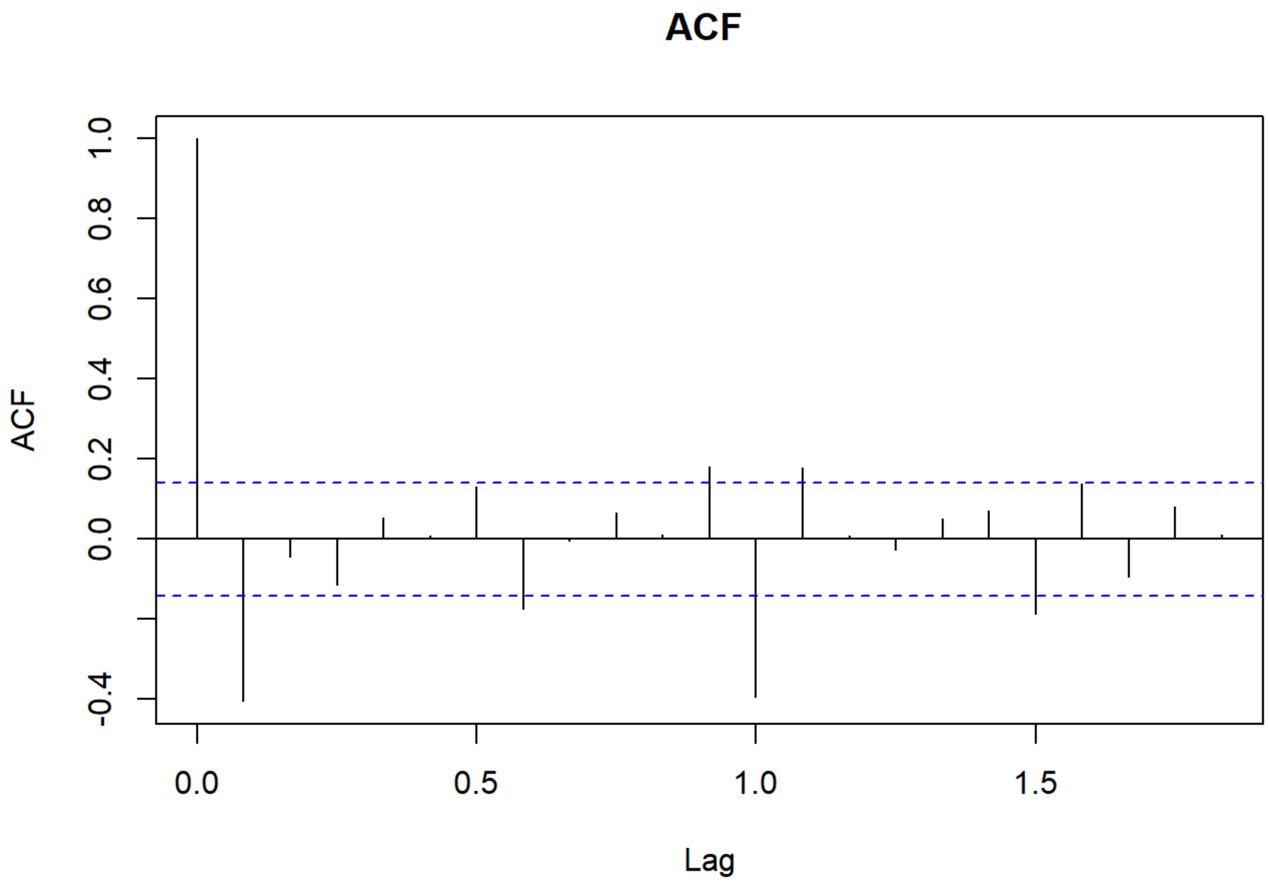


Étape d'identification: vérifions la stationnarité des résidus avec un test de **Dickey-Fuller** et affichons les fonctions d'autocorrélation et d'autocorrélation partielle.

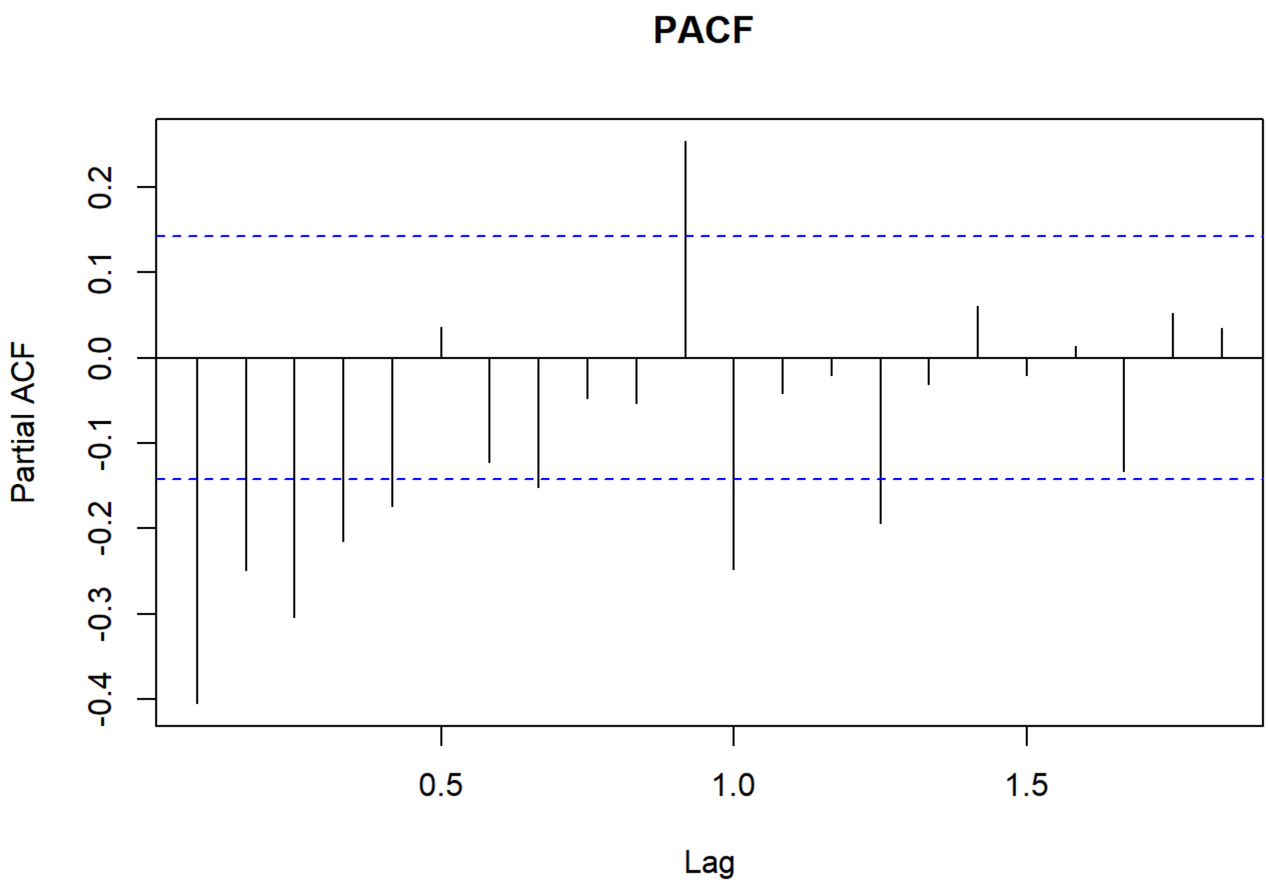
```
adf.test(diff(diff(d_train,lag=1,difference=1),lag=12,difference=1)) # Test de Dickey-Fuller (stationnarité)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(diff(d_train, lag = 1, difference = 1), lag = 12, difference = 1)
## Dickey-Fuller = -8.0407, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
acf(diff(diff(d_train,lag=1,difference=1),lag=12,difference=1), main = "ACF ")
```




```
pacf(diff(diff(d_train,lag=1,difference=1),lag=12,difference=1), main = "PACF ")
```

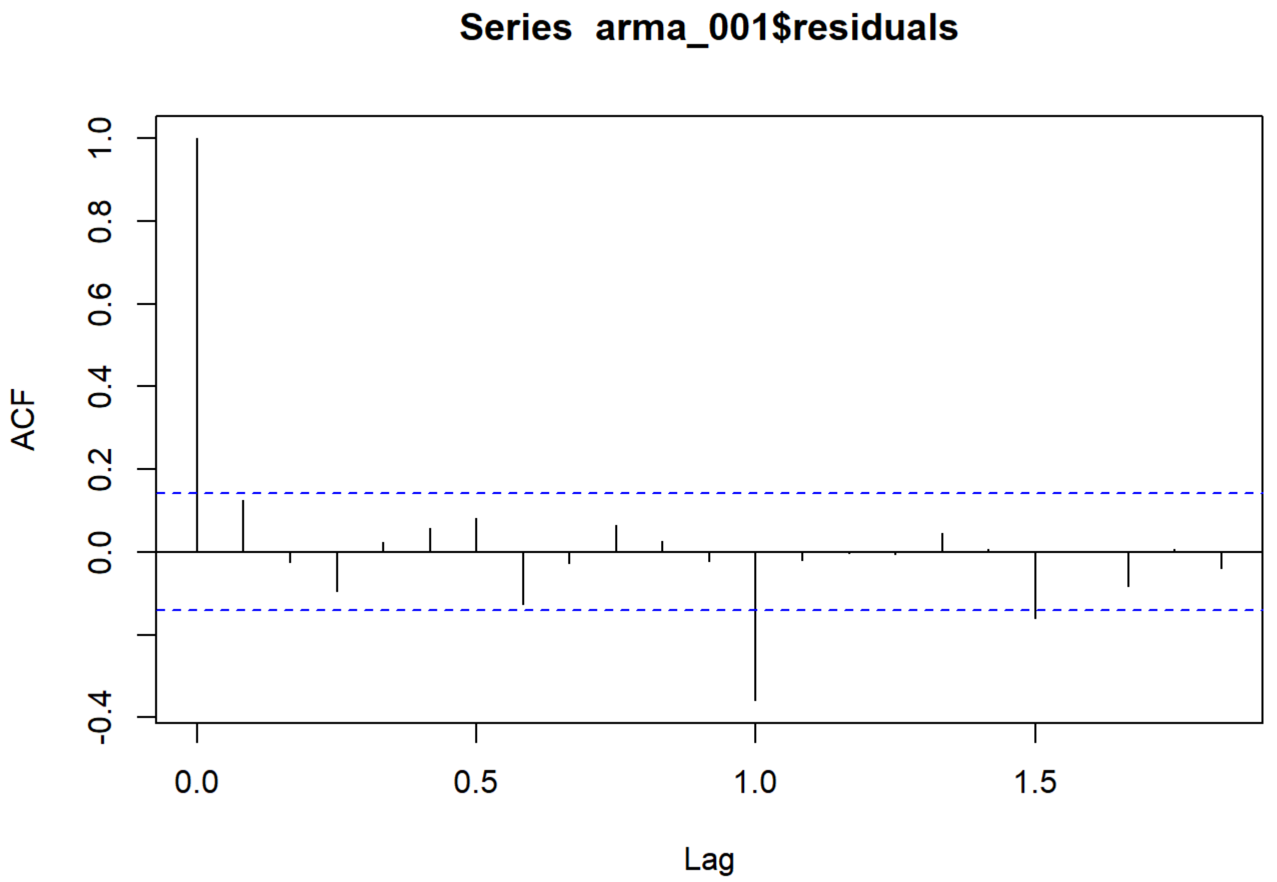


Le test de **Dickey-Fuller** ne nous permet pas de rejeter l'hypothèse de stationnarité de la série différenciée. L'ACF est presque nul à l'ordre 2 et on observe une décroissance exponentielle du PACF avec oscillation. On peut supposer un processus de type MA(1) en première approche.

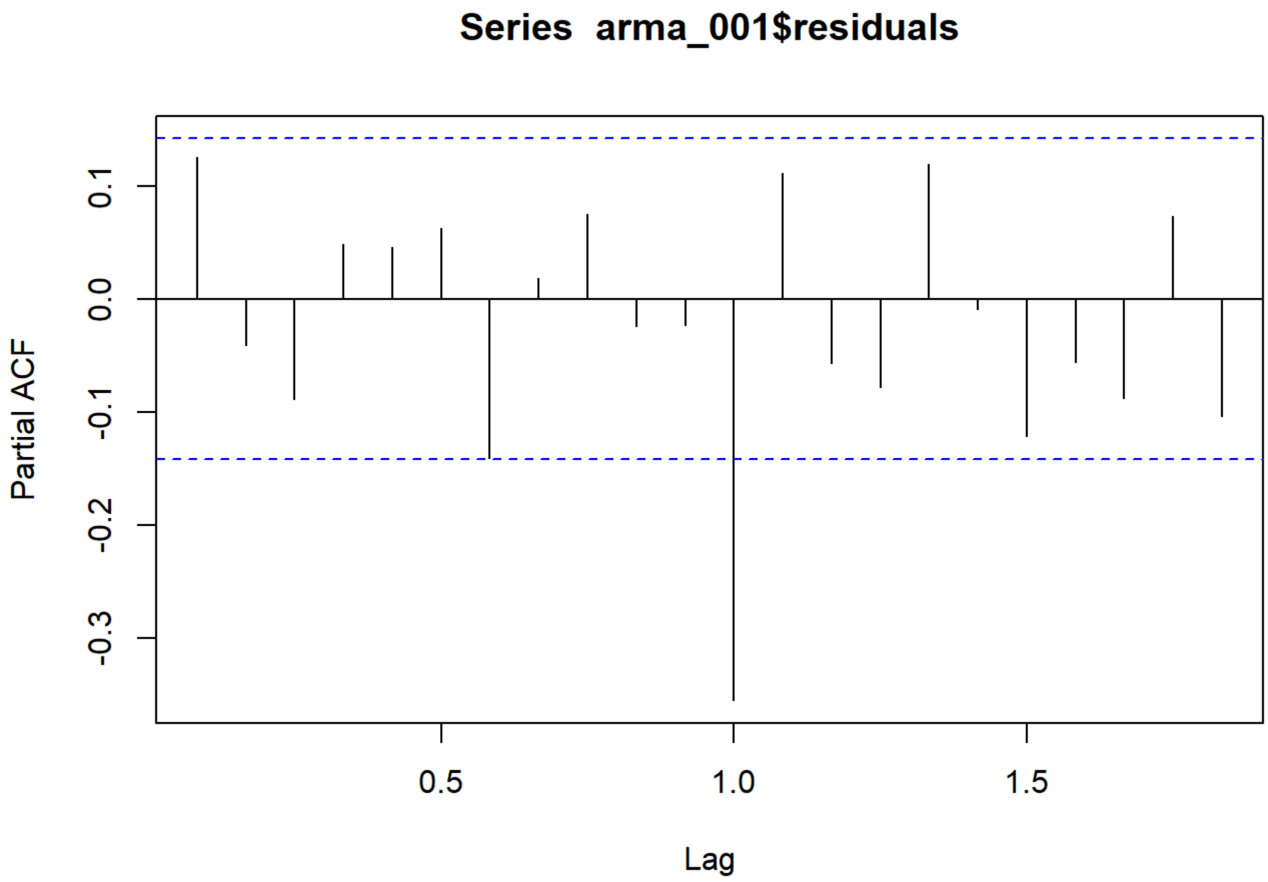
```
arma_001 <- arima(diff(diff(d_train,lag=1,difference=1),lag=12,difference=1), order=c(0,0,1))
Box.test(arma_001$residuals,lag=20,type="Ljung-Box") # Test de blancheur des résidus.
```

```
##
##  Box-Ljung test
##
## data:  arma_001$residuals
## X-squared = 45.435, df = 20, p-value = 0.0009631
```

```
acf(arma_001$residuals)
```



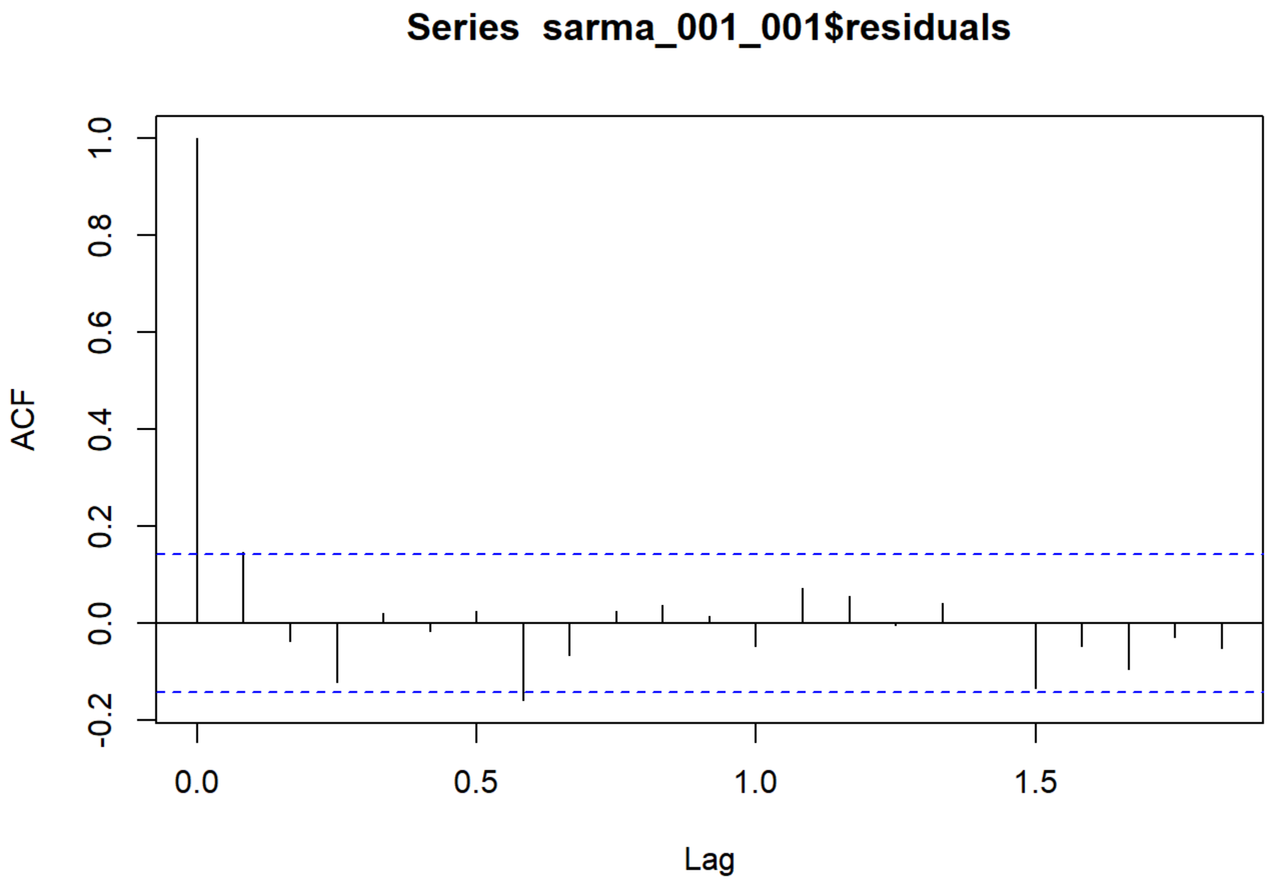

```
pacf(arma_001$residuals)
```



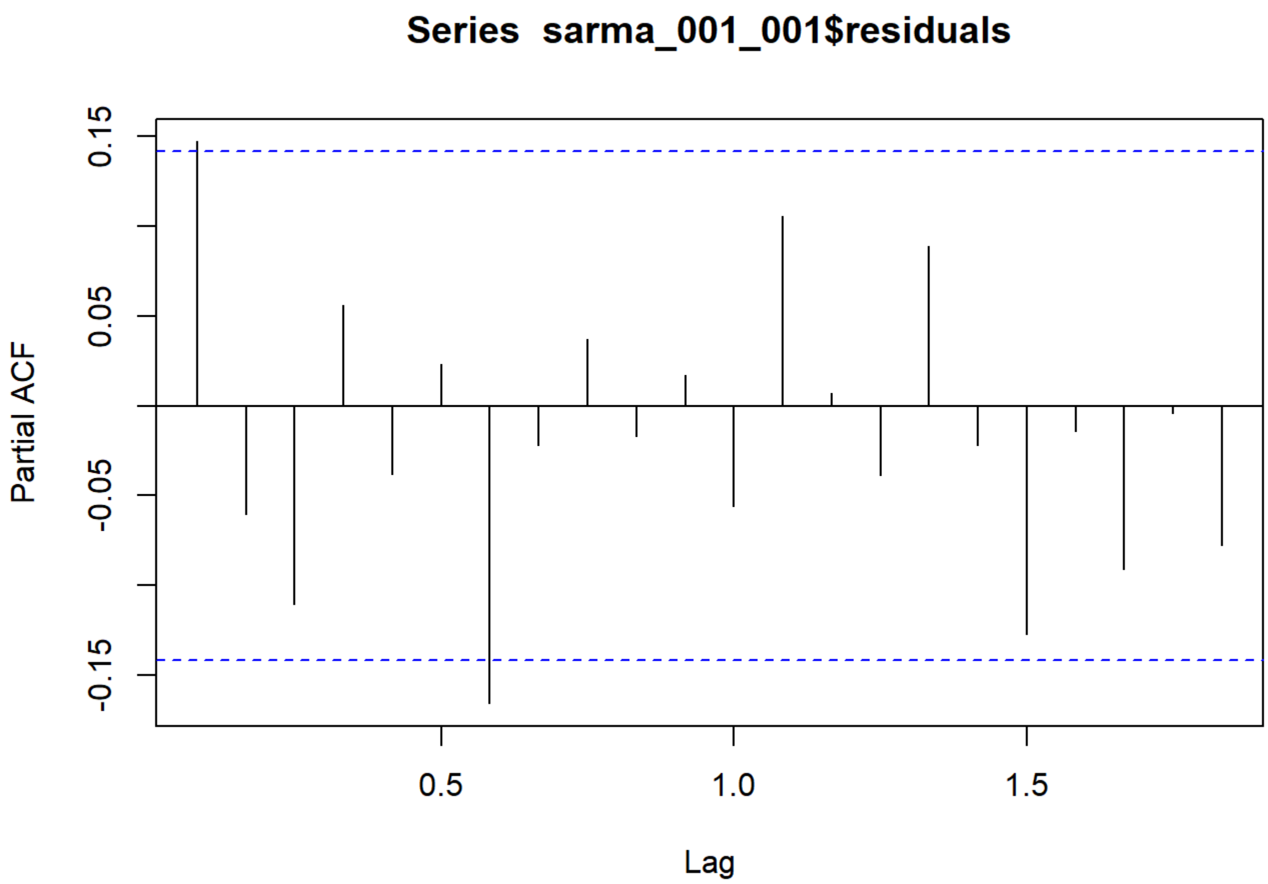
D’après les résultats du test on peut rejeter la blancheur des résidus.
C’est confirmé par les graphiques des ACF et PACF des résidus du modèle.
On peut supposer qu’il reste encore de la saisonnalité au vu du pic du 12e ordre sur les deux graphiques.

Les tests de plusieurs modèles ARMA avec d’autres valeurs de p et q n’ont rien donné non plus (non représenté ici).
Nous envisageons un modèle SARMA pour prendre en compte la saisonnalité restante dans les résidus (ordre 12 à priori).

```
sarma_001_001<- Arima(diff(diff(d_train,lag=1,difference=1),lag=12,difference=1), order=c(0,
0,1), seasonal=list(order=c(0,0,1),period=12))
acf(sarma_001_001$residuals)
```




```
pacf(sarma_001_001$residuals)
```



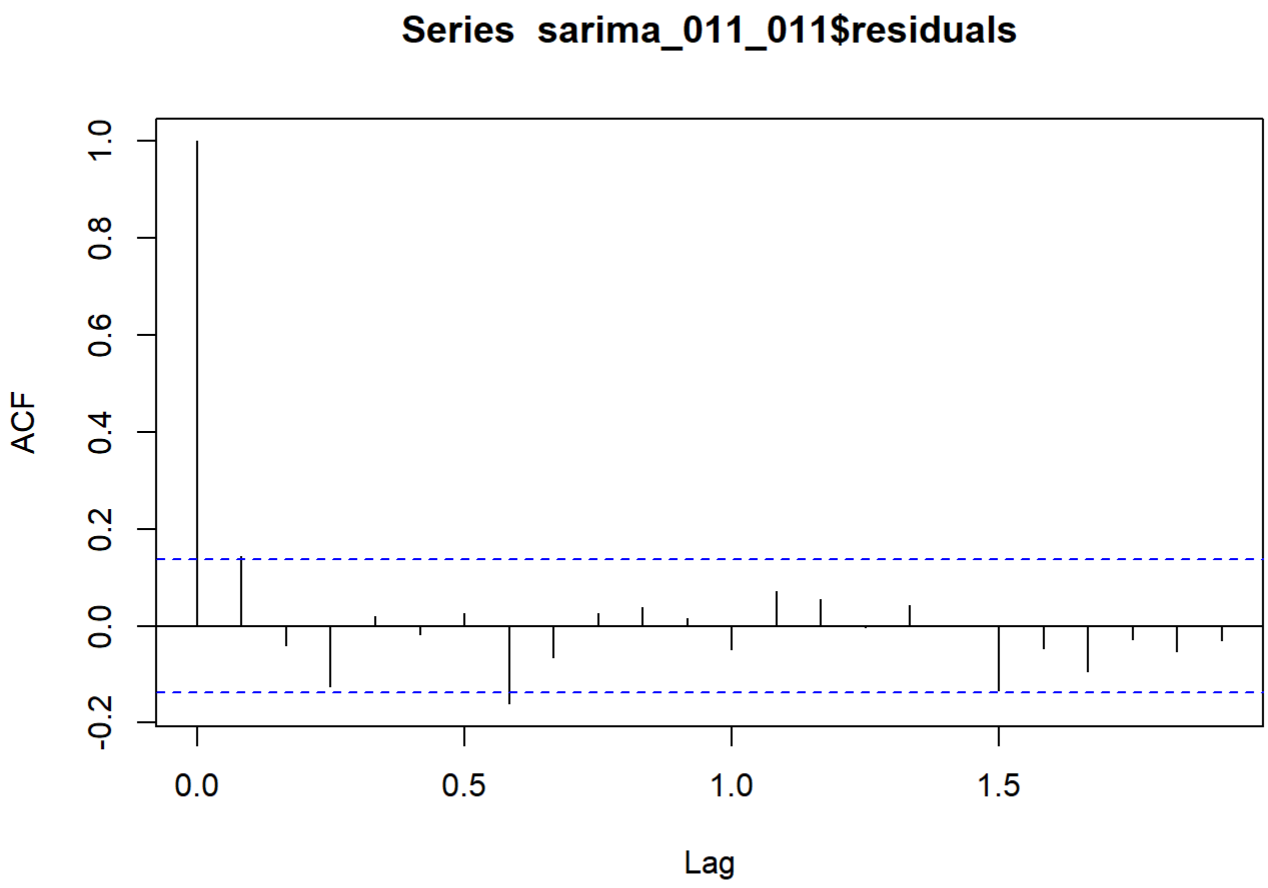
```
Box.test(sarma_001_001$residuals,lag=20,type="Box-Pierce")
```

```
##
##  Box-Pierce test
##
## data:  sarma_001_001$residuals
## X-squared = 21.481, df = 20, p-value = 0.3693
```

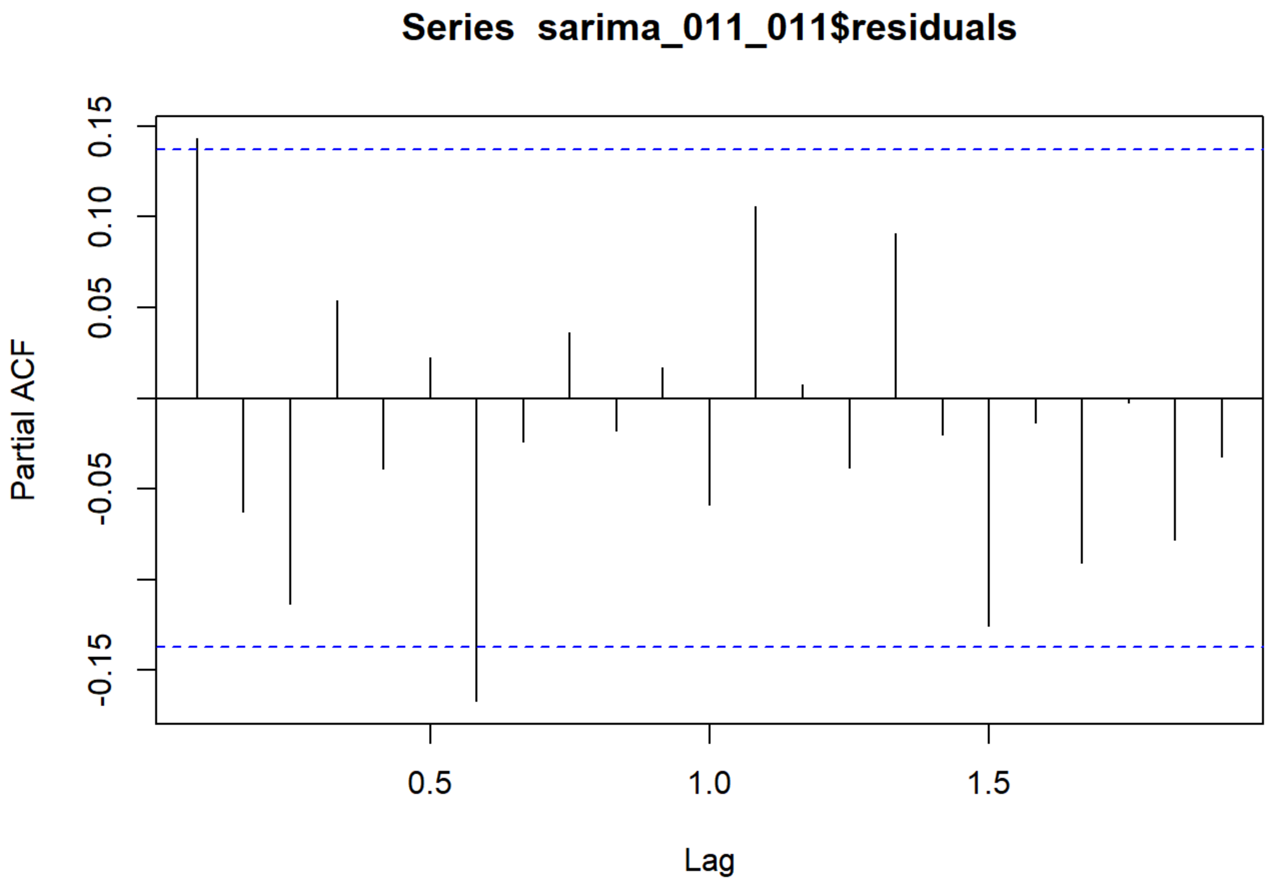
Cette fois, le test ne permet pas de rejeter l'hypothèse de blancheur.
Les graphiques des ACF et PACF sont acceptables.

Partons de ce modèle SARMA (**sarma_001_001**), et analysons-le plus précisément.
Pour commencer, nous pouvons simplifier la notation et recalculer notre modèle comme un SARIMA (**sarima_011_011**):

```
sarima_011_011<- Arima(d_train,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=12))
acf(sarima_011_011$residuals)
```




```
pacf(sarima_011_011$residuals)
```




```
Box.test(sarima_011_011$residuals,lag=20,type="Box-Pierce") # Test de blancheur des résidus.
```

```
##
##  Box-Pierce test
##
## data:  sarima_011_011$residuals
## X-squared = 22.884, df = 20, p-value = 0.2945
```

Remarque: nous ne voyons pas d'explication au fait que le résultat du test de blancheur du modèle **sarma_001_001** sur la série déjà stationnarisée soit différent du résultat de celui du modèle **sarima_011_011** sur la série “brute” (respectivement 0.3693 et 0.2945). Les deux modèles sont censés être pourtant mathématiquement identiques (sans doute une sous-couche de la fonction crée-t-elle une légère différence entre eux).
Par contre, les graphiques ACF et PACF sont bien équivalents pour les deux modèles.
Comme on peut le voir ci-dessous, les coefficients et les indicateurs AIC, AICc, BIC sont, eux, presque égaux:

```
sarma_001_001
```

```
## Series: diff(diff(d_train, lag = 1, difference = 1), lag = 12, difference = 1)
## ARIMA(0,0,1)(0,0,1)[12] with non-zero mean
##
## Coefficients:
##           ma1     sma1    mean
##       -0.8387  -0.4819  0.4165
## s.e.   0.0501   0.0651  0.8166
##
## sigma^2 = 15268:  log likelihood = -1191.76
## AIC=2391.52   AICc=2391.74   BIC=2404.53
```

```
sarima_011_011
```

```
## Series: d_train
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##           ma1     sma1
##       -0.8341  -0.4788
## s.e.   0.0493   0.0648
##
## sigma^2 = 15214:  log likelihood = -1191.89
## AIC=2389.77   AICc=2389.9   BIC=2399.53
```

Nous retenons notre modèle **sarima_011_011** pour sa simplicité d'écriture.

```
round(cor.arma(sarima_011_011), 2)
```

```
##        ma1  sma1
## ma1   1.00 -0.09
## sma1 -0.09  1.00
```

```
t_stat(sarima_011_011)
```

```
##              ma1      sma1
## t.stat -16.90994 -7.388764
## p.val    0.00000  0.000000
```

Pas de problème de colinéarité et les *p-values* sont proches de 0.
Notre modèle est acceptable.

Générons un modèle concurrent automatiquement via la fonction *auto.arima()*:

```
model_auto <- auto.arima(d_train)
summary (model_auto)
```

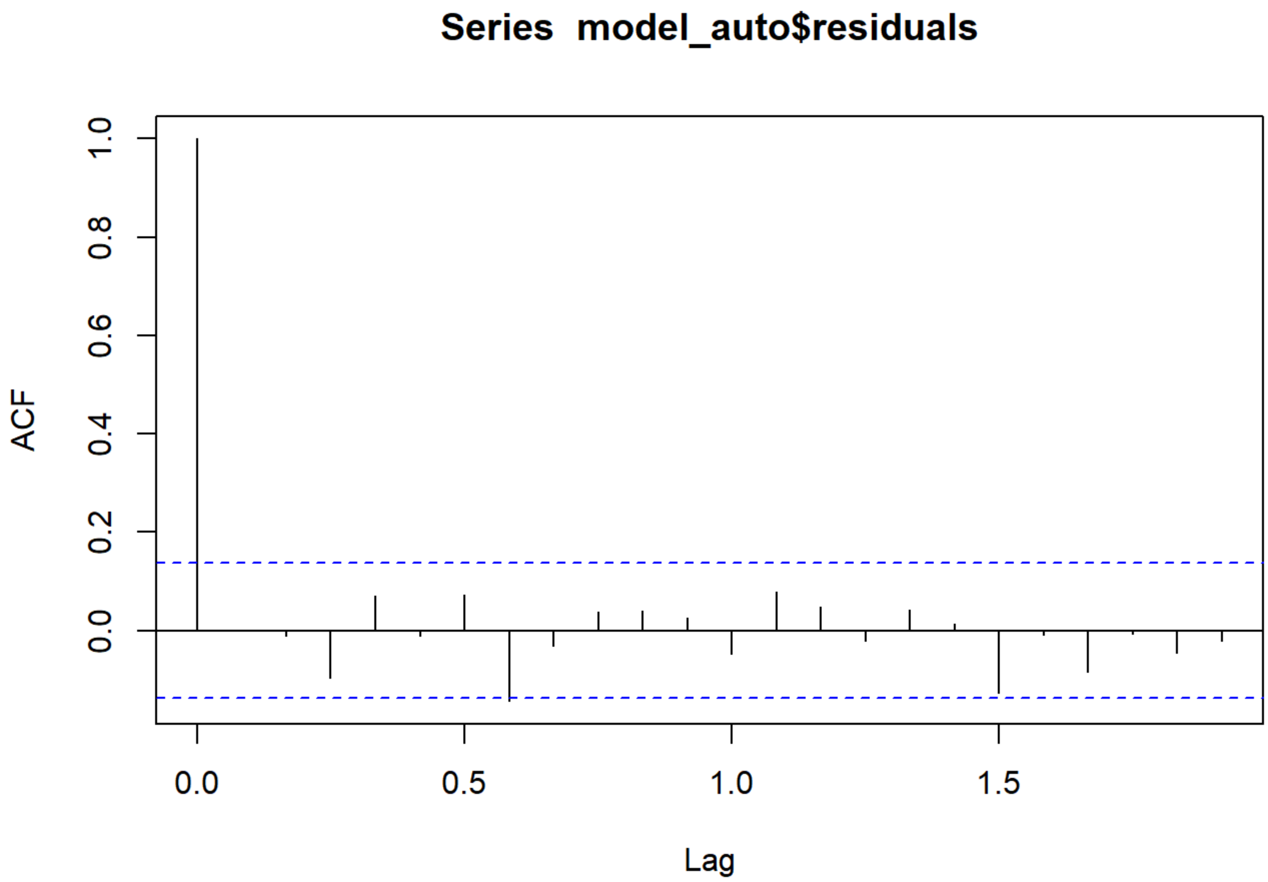
```
## Series: d_train
## ARIMA(1,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.2021  -0.8907  -0.4985
## s.e.  0.0828   0.0375   0.0669
##
## sigma^2 = 14801:  log likelihood = -1189
## AIC=2386   AICc=2386.21   BIC=2399.01
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 5.398044 116.7929 82.36166 0.2528859 3.286532 0.6966886
##                      ACF1
## Training set 0.0004493165
```

Par rapport à notre modèle “fait à la main”, la fonction *auto.arima()* retient en plus une composante autorégressive d’ordre 1.
Testons la blancheur des résidus:

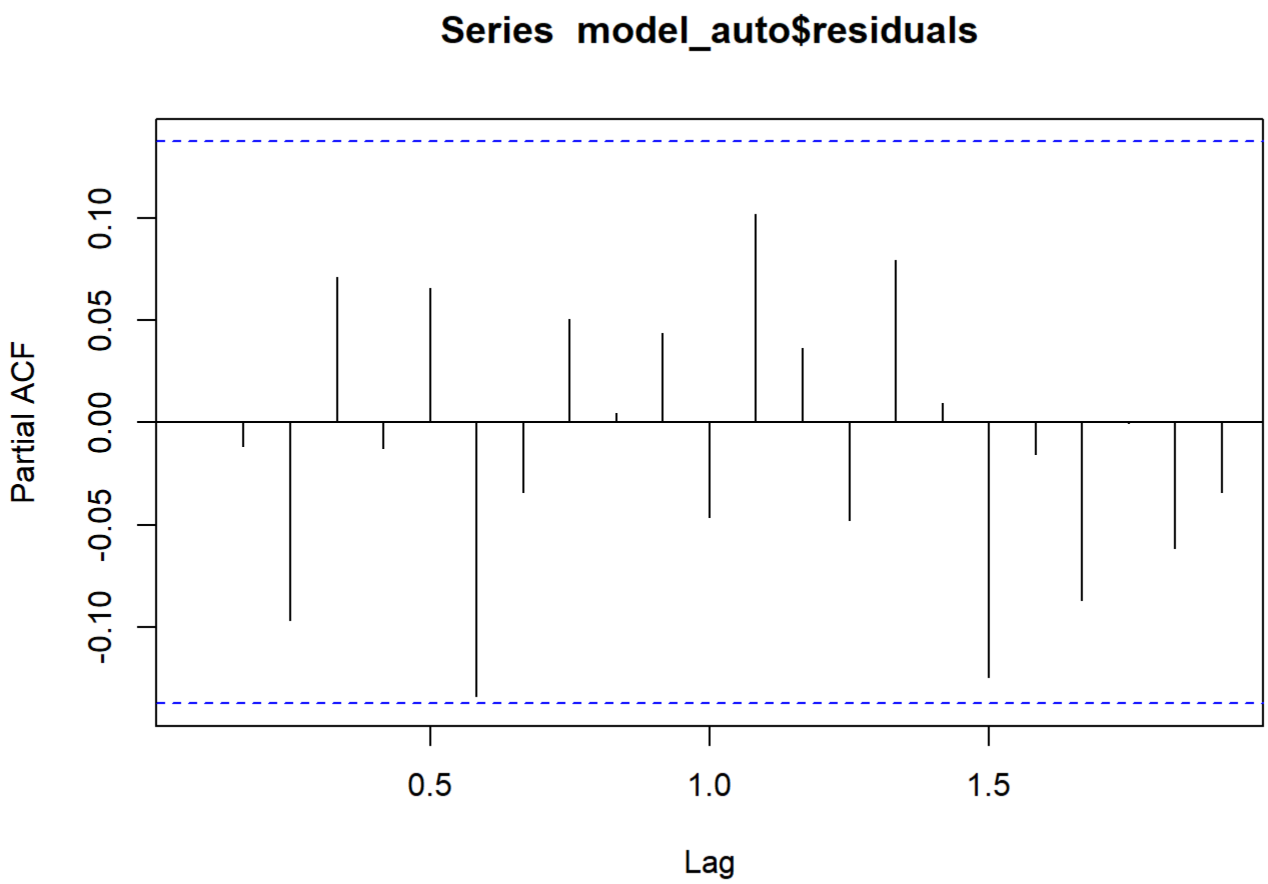
```
Box.test(model_auto$residuals,lag=20,type="Box-Pierce")
```

```
##
##  Box-Pierce test
##
## data:  model_auto$residuals
## X-squared = 16.639, df = 20, p-value = 0.6762
```

```
acf(model_auto$residuals)
```




```
pacf(model_auto$residuals)
```



C’est concluant. Notons que la *p-value* est plus élevée que celle de notre modèle, respectivement 0.6762 contre 0.2945.
L’AICC est également meilleure: 2386.21 pour le modèle auto contre 2389.9 pour notre modèle (la différence est cependant minime).
Testons la colinéarité et la significativité des coefficients:

```
round(cor.arma(model_auto), 2)
```

```
##        ar1   ma1  sma1
## ar1   1.00 -0.49 -0.12
## ma1  -0.49  1.00 -0.09
## sma1 -0.12 -0.09  1.00
```

```
t_stat(model_auto)
```

```
##             ar1       ma1      sma1
## t.stat 2.441972 -23.74509 -7.446865
## p.val  0.014607   0.00000  0.000000
```

Pas de colinéarité critique (>90) et les p-values sont proches de 0.

Les 2 modèles sont très proches, mais par exigence de parcimonie, nous retenons notre modèle “fait à la main” **sarima_011_011** comme modèle pour nos prévisions.

Par curiosité, essayons un autre concurrent avec la fonction *armaselect()*.
Notons que cette fonction peut sélectionner exclusivement des modèles de type ARMA (ordres p et q uniqement), nous lui fournissons donc une série déjà stationnarisée.
Nous ajoutons la contrainte suivante: les valeurs de p et q ne peuvent pas excéder 3. Nos essais sans contraintes ont, de manière suprenante, donné des modèles uniquement avec des composantes AR (le premier modèle retenu était p=12 et q=0).

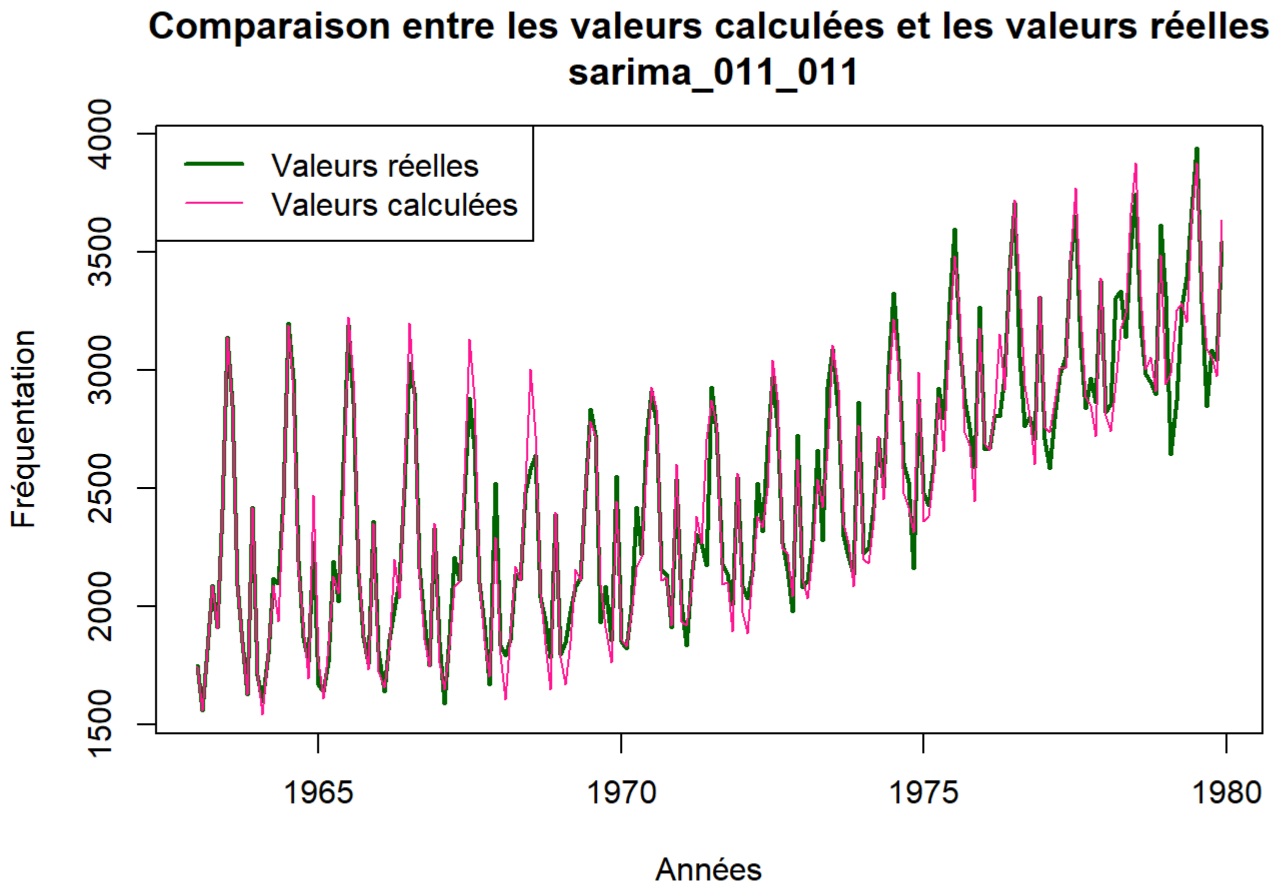
```
armaselect(diff(diff(d_train,lag=1,difference=1),lag=12,difference=1), max.p = 3, max.q = 3, nbmod = 10)
```

```
##       p q      sbc
##  [1,] 3 0 1913.146
##  [2,] 2 0 1925.477
##  [3,] 2 3 1927.175
##  [4,] 3 1 1927.531
##  [5,] 1 2 1928.213
##  [6,] 1 3 1928.227
##  [7,] 3 2 1929.622
##  [8,] 3 3 1930.400
##  [9,] 1 0 1931.785
## [10,] 0 1 1932.613
```

Malgré les contraintes, la fonction *armaselect()* favorise les modèles avec des ordres AR importants.
Nous avons vérifié: ils n’obtiennent pas de meilleurs résultats que nos modèles étudiés jusqu’ici.
Nous pouvons en déduire que la fonciton *armaselect()* n’est pas adaptée aux séries à la saisonnalité trop complexe (i.e.: comme on l’a vu, il reste de la saisonnalité dans la série pourtant differenciée).

Comparons graphiquement les valeurs calculées par notre modèle **sarima_011_011** et les valeurs réelles:

```
plot(d_train, type = "l", col = "darkgreen", lwd= 2,main = "Comparaison entre les valeurs cal
culées et les valeurs réelles\n sarima_011_011", ylab="Fréquentation", xlab = "Années")
lines(fitted(sarima_011_011), col = "deeppink", lwd= 1)
legend("topleft", legend = c("Valeurs réelles", "Valeurs calculées"), col = c("darkgreen", "d
eeppink"), lwd = c(2,1))
```



Nous constatons que notre modèle prévoit mal la baisse de variance saisonnière entre 1965 et 1970. On note aussi que le décrochage (inattendu) du printemps 1979 n'est logiquement pas prévu. Par contre, les deux courbes se rapprochent en début d'hiver, fin 1979. Remarque: le modèle **model_auto** (non représenté ici) calculé avec la fonction *auto.arima()* ne donne aucune différence perceptible avec notre **sarima_011_011** .

Approche par lissage exponentiel

Nous basant sur la série décomposée affichée plus haut, nous proposons en première approche un modèle **"A,A,A"**. Cependant, comme déjà évoqué, ce choix est discutable en fonction de la période que nous privilégions (sur les dix premières années, un modèle multiplicatif aurait sans doute été plus judicieux, de type "M,A,M" à priori). Nous favorisons logiquement les dernières années.

```
model_lissage_AAA = ets(d_train, model="AAA")
summary (model_lissage_AAA)
```

```
## ETS(A,A,A)
##
## Call:
##  ets(y = d_train, 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
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set 6.372573 126.3753 91.91021 0.1744532 3.681543 0.7774589 0.2434013
```

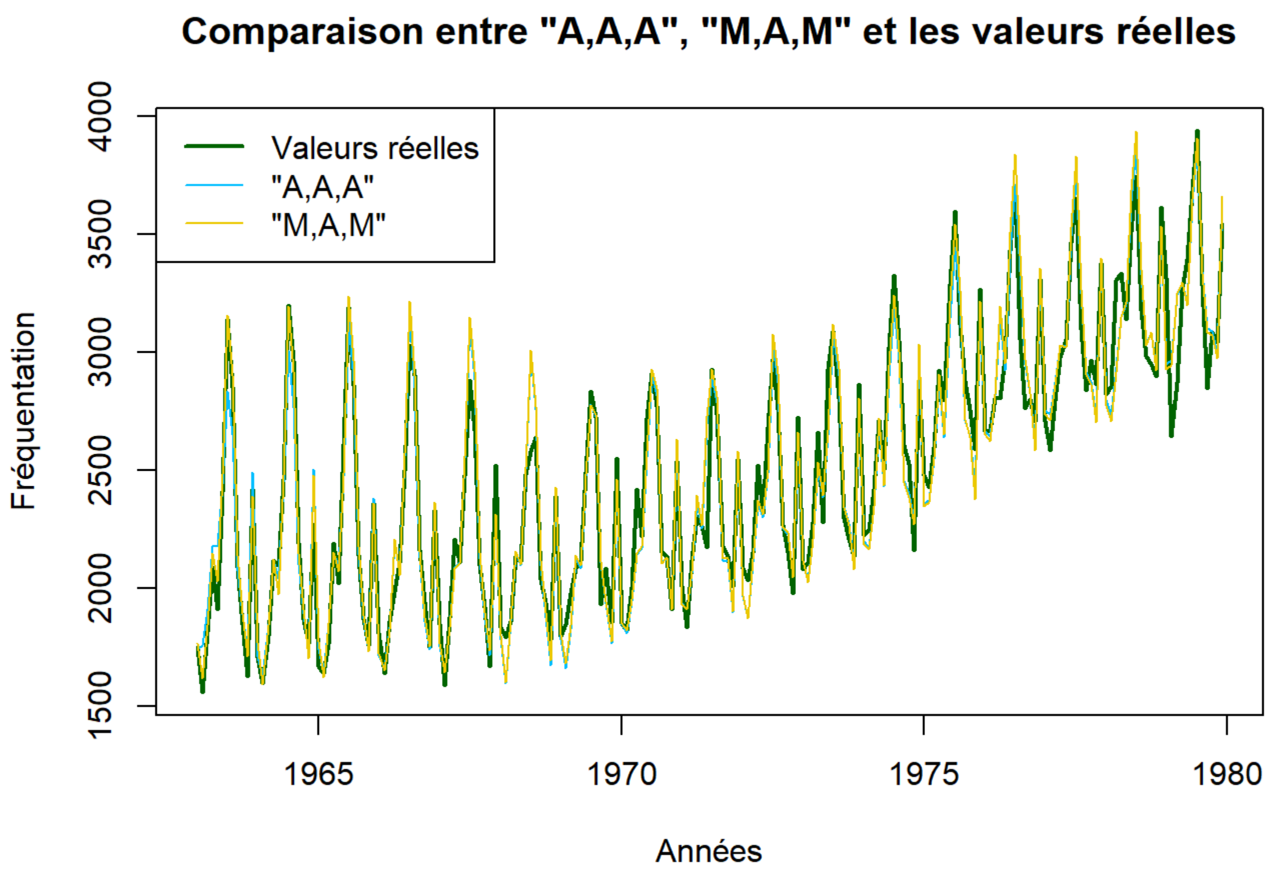
Élaborons un modèle concurrent via la fonction de sélection automatique *ets()* (selon le critère AICc):

```
model_lissage_auto = ets(d_train, ic="aicc")
summary (model_lissage_auto)
```

```
## ETS(M,A,M)
##
## Call:
##  ets(y = d_train, ic = "aicc")
##
##   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 0.2187159
```

Le modèle retenu est celui à erreur multiplicative, tendance additive et saisonnalité multiplicative (**“M,A,M”**). Ses trois critères AIC, AICc et BIC sont meilleurs que ceux de notre modèle “fait à la main”.

```
plot(d_train, type = "l", col = "darkgreen", lwd= 2, main = 'Comparaison entre "A,A,A", "M,A,M" et les valeurs réelles', ylab="Fréquentation", xlab = "Années")
lines(fitted(model_lissage_AAA), type = "l", col="deepskyblue", xlab = "Temps", ylab = "Valeurs ajustées")
lines(fitted(model_lissage_auto), type = "l", col="gold2", xlab = "Temps", ylab = "Valeurs ajustées")
legend("topleft", legend = c("Valeurs réelles", '"A,A,A"',  '"M,A,M"'), col = c("darkgreen", "deepskyblue", "gold2"), lwd = c(2,1,1))
```



On peut observer que notre modèle **“A,A,A”** fait un peu mieux que le modèle sélectionné automatiquement **“M,A,M”**, sauf pour les trois premiers pics qui sont mieux suivis par le modèle multiplicatif (sans surprise, la série étant de type multiplicatif sur cette période).
Globalement, les différences restent cependant minimes.
Nous avons testé les deux modèles en les poussant hors des limites du raisonnable, en imposant des prédictions sur 15 ans (non représenté ici): les valeurs prédites restent assez proches d’un modèle à l’autre. Par contre, l’intervalle de confiance du modèle **“M,A,M”** devient rapidement beaucoup plus large que celui de son concurrent.

Pour la suite, nous conservons le modèle **“A,A,A”**.

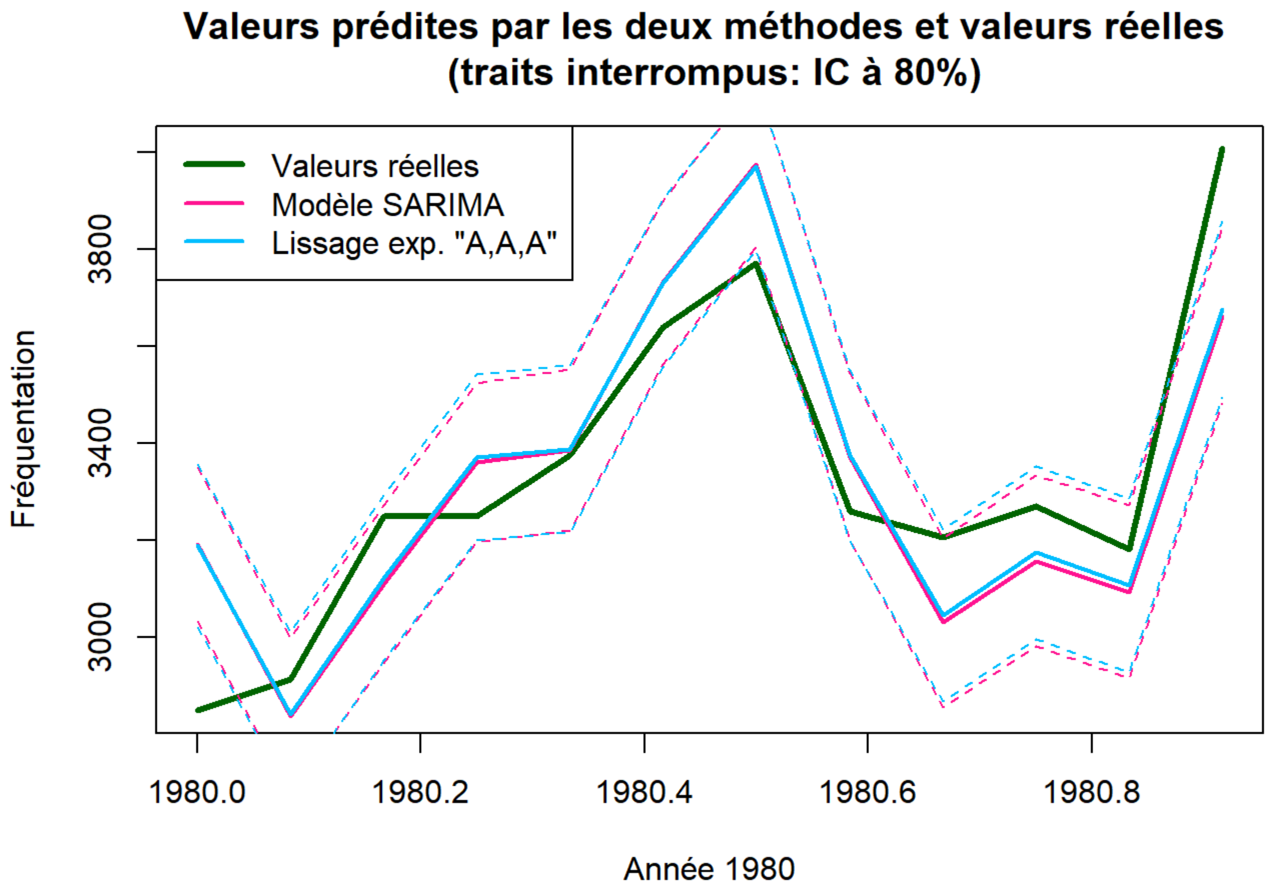
Comparaisons entre le modèle SARIMA et le modèle par lissage exponentiel

Comparons les prédictions des deux méthodes:

```
##modèle SARIMA_011_011
# Prédictions et intervalles de confiance
pred_SARIMA <- forecast(sarima_011_011, h = 12, level = 80)
# Extraction des valeurs prédites et les intervalles de confiance inférieurs et supérieurs
predicted_values_SARIMA <- pred_SARIMA$mean  # extrait valeurs prédites moyennes SARIMA
lower_ci_SARIMA <- pred_SARIMA$lower  # extrait intervalle de confiance inférieur SARIMA
upper_ci_SARIMA <- pred_SARIMA$upper  # extrait intervalle de confiance supérieur SARIMA

##Lissage exponentiel
# Prédictions et intervalles de confiance
pred_lissage <- forecast(model_lissage_AAA,h=12, level=80)
# Extraction des valeurs prédites et les intervalles de confiance inférieurs et supérieurs
predicted_values_lissage <- pred_lissage$mean     # extrait valeurs prédites moyennes lissage exponentiel
lower_ci_lissage <- pred_lissage$lower  # extrait intervalle de confiance inférieur lissage exponentiel
upper_ci_lissage <- pred_lissage$upper # extrait intervalle de confiance supérieur lissage exponentiel

## Graphique valeurs réelles, prédites et intervalles de confiance
plot(d_test, type = "l", main = "Valeurs prédites par les deux méthodes et valeurs réelles \n(traits interrompus: IC à 80%)", col= "darkgreen",lwd= 3, ylab="Fréquentation", xlab = "Année 1980")
lines(predicted_values_SARIMA, col = "deeppink",lwd= 2) # affiche valeurs moyennes SARIMA
lines(lower_ci_SARIMA, col = "deeppink", lty = 2)  # affiche intervalle de confiance inférieur SARIMA
lines(upper_ci_SARIMA, col = "deeppink", lty = 2)  # affiche intervalle de confiance supérieur SARIMA
lines(predicted_values_lissage, col = "deepskyblue",lwd= 2) # affiche valeurs moyennes lissage exp.
lines(lower_ci_lissage, col = "deepskyblue", lty = 2)  # affiche intervalle de confiance inférieur lissage exp.
lines(upper_ci_lissage, col = "deepskyblue", lty = 2)  # affiche intervalle de confiance supérieur lissage exp.
legend("topleft", legend = c("Valeurs réelles", "Modèle SARIMA", 'Lissage exp. "A,A,A"'), col = c("darkgreen", "deeppink", "deepskyblue"), lty = c(1, 1, 1), lwd = c(3,2,2))
```



Nous obtenons presque les mêmes résultats avec les deux méthodes, aussi bien quant aux valeurs prédites que pour les intervalles de confiance.
Le pic de l'été est surestimé par nos modèles par rapport aux valeurs réelles. Par contre, le pic de fin d'année (on n'en voit ici que le début: le mois de décembre) est, lui, sous-estimé.
La courbe des valeurs réelles est à trois reprises en dehors du "couloir" des intervalles de confiance: tout au début de l'année, au plus haut du pic d'été et les deux derniers mois de l'année.

Autre approche en partant de 1970

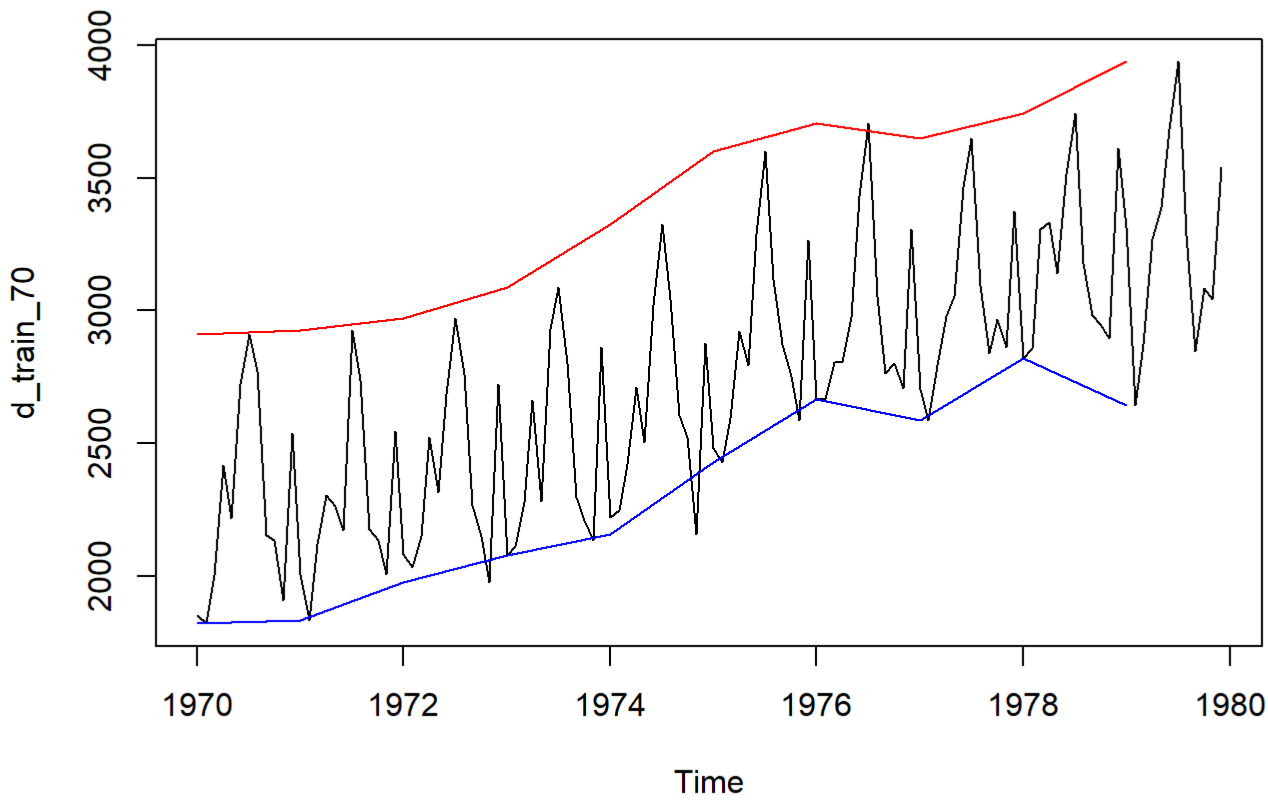
Comme nous l'avons déjà noté, la série sur laquelle nous avons travaillé jusqu'ici (de 1963 à 1979) évolue en comportement sur la durée.
Une autre approche consisterait à ne retenir les valeurs qu'à partir de 1970, date à laquelle la série adopte un comportement plus régulier: tendance croissante de type à priori additif et saisonnalité également de type additif.

Vérifions que la saisonnalité est bien de type additif par la méthode de la bande:

```
# Création d'une nouvelle variable ts "d_train_70" de 1970 à 1979
d_train_70 <- window(d_train, start = c(1970, 1))

MatX=matrix(data=d_train_70,nrow=12)
Min=apply(MatX,2,min)
Max=apply(MatX,2,max)
AnneeMin=c(1970:1979)
AnneeMax=c(1970:1979)

plot(d_train_70)
lines(AnneeMin,Min,col="blue",type = "l") # bande inférieure
lines(AnneeMax,Max,col="red",type = "l")  # bande supérieure
```



La saisonnalité est bien, globalement, de type additif.

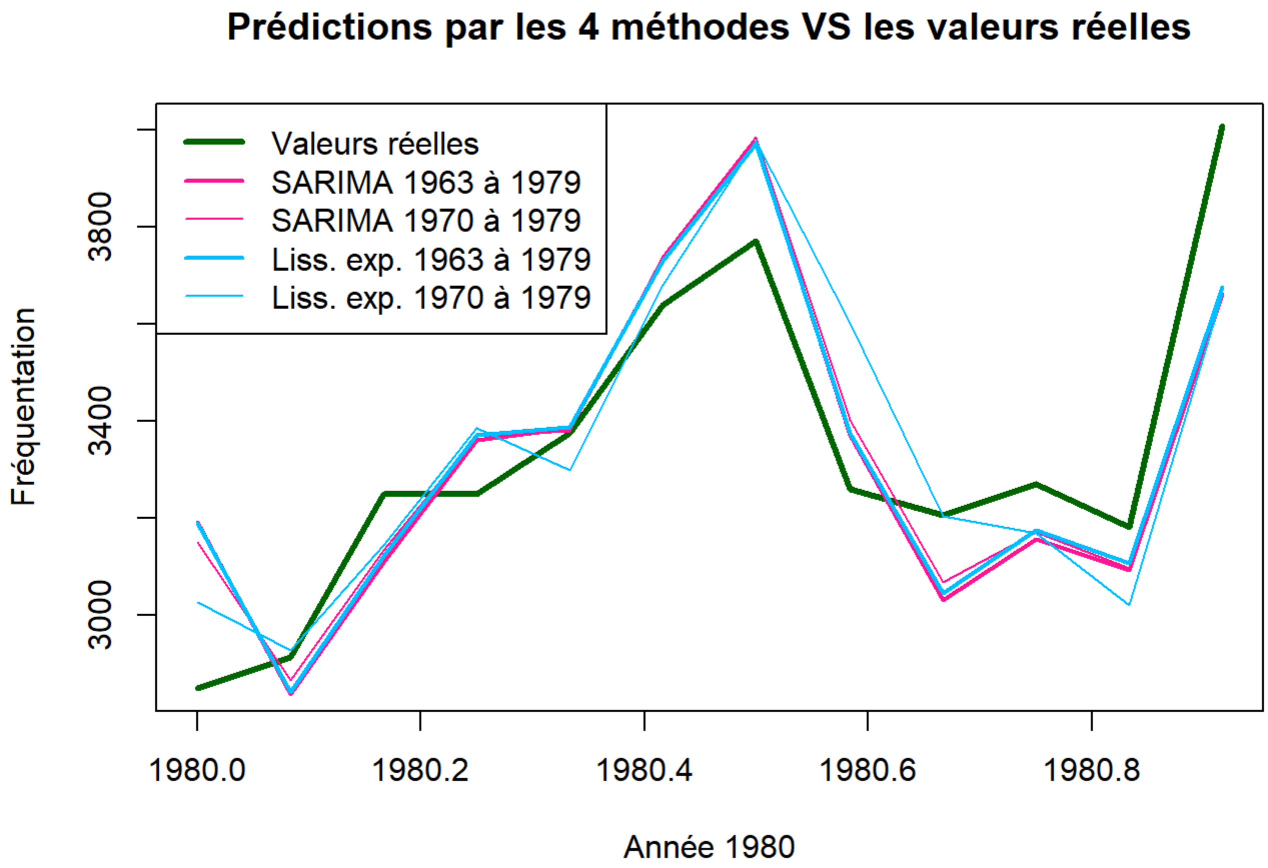
Nous ne détaillons pas ici les démarches pour aboutir aux modèles à partir de l'année 1970. Ce sont les mêmes étapes que précédemment: différenciation, test de stationnarité, ACF, PACF, essai modèle, test de blancheur des résidus, test de colinéarité et de significativité des coefficients, etc.
Les modèles retenus sont:

- le **sarima_70_111_011**, un ARIMA(1,1,1)(0,1,1)[12]
- le **model_lissage_70_AAA**, un modèle par lissage exponentiel de type **"A,A,A"**

Notons que la fonction *auto.arima()* n'a pas donné de résultats probants en proposant un modèle ARIMA(1,0,0)(0,1,1)[12]. Ce modèle ne retient donc pas de composante de différenciation dans la partie non saisonnière.
Création des modèles:

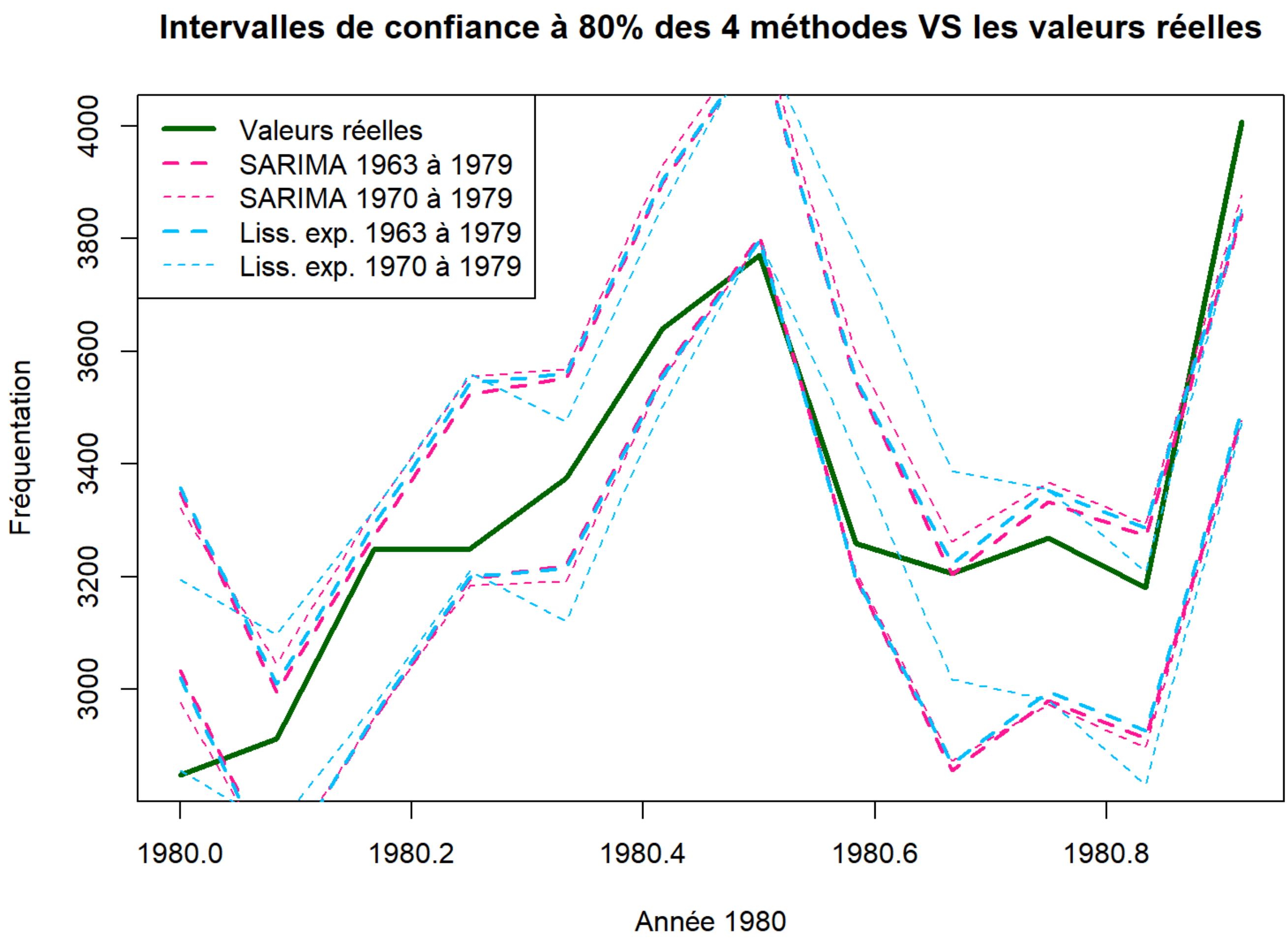
```
sarima_70_111_011<- Arima(d_train_70,order=c(1,1,1),seasonal=list(order=c(0,1,1),period=12))
model_lissage_70_AAA = ets(d_train_70, model="AAA", damped = FALSE)
```

Comparons les prédictions des 4 modèles:
(Nous n'affichons pas ici le code qui est identique aux codes précédents dans sa structure.)



Concernant les modèles SARIMA, les courbes des valeurs prédites sont presque identiques, que le modèle soit calculé sur l'ensemble des valeurs ou à partir de 1970.
La méthode par lissage exponentiel montre un peu plus de différences entre les deux cas. Notre approche initiale consistant à utiliser l'ensemble des données donne ici de meilleurs résultats.

Faisons la même comparaison mais cette fois avec les intervalles de confiance à 80%:



L'intervalle de confiance à 80% du modèle par lissage exponentiel n'utilisant les valeurs qu'à partir de 1970 encadre moins bien les valeurs réelles que les trois autres modèles qui obtiennent, eux, des résultats très proches.

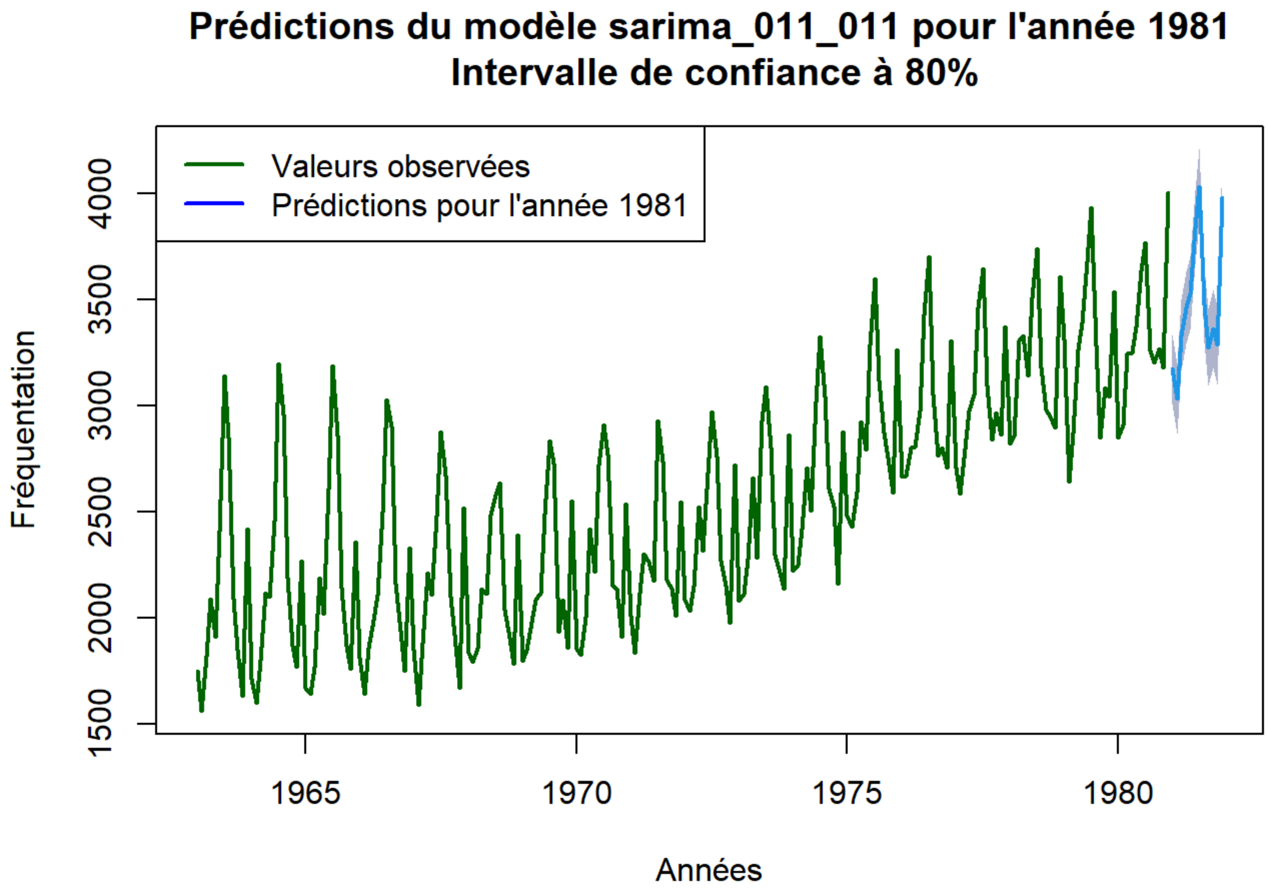
Prévisions pour l'année 1981

Nous choisissons d'utiliser pour prédire les valeurs de l'année 1981 le modèle **sarima_011_011** (sur les données à partir de 1963) recalculé pour tenir compte de l'année 1980. Notons que les modèles **sarima_70_111_011** et **model_lissage_AAA** pourraient à priori tout aussi bien convenir.

```
sarima_011_011<- Arima(d,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=12))
pred_81=forecast(sarima_011_011, h=12, level =80)

plot(pred_81, col="darkgreen",main = "Prédictions du modèle sarima_011_011 pour l'année 1981
\n Intervalle de confiance à 80%", lwd=2, ylab="Fréquentation", xlab = "Années")

legend("topleft",c("Valeurs observées","Prédictions pour l'année 1981"),col=c("darkgreen","bl
ue"),lty=rep(1,2),lwd = c(2,2))
```



Conclusion

Les deux approches, SARIMA et lissage exponentiel, arrivent à des résultats comparables.
La littérature est vaste quant aux avantages de l'une ou l'autre par rapport à son concurrent. Il n'y a pas unanimité à ce sujet, mais pour résumer, les résultats sont toujours très proches, à condition de bien paramétrer son modèle, et c'est en fonction de la série à analyser que le choix doit s'opérer. Cependant, les modèles de type SARIMA semblent dominer le marché, notamment quand il s'agit d'étudier des évolutions à plus long terme.