Approche de Box-Jenkins pour l'identification et l'estimation des modèles ARIMA -Partie pratique-

Yassine HALLAL

2023-01-29

Contents

1. Données :

Dans cet exemple, nous allons essayer d'appliquer le modèle ARIMA sur les données des cours quotidiens du "*MSCI Emerging Markets ETF" ou (***EEM***)*, de *Janvier 2021* jusqu'à *décembre 2022.*

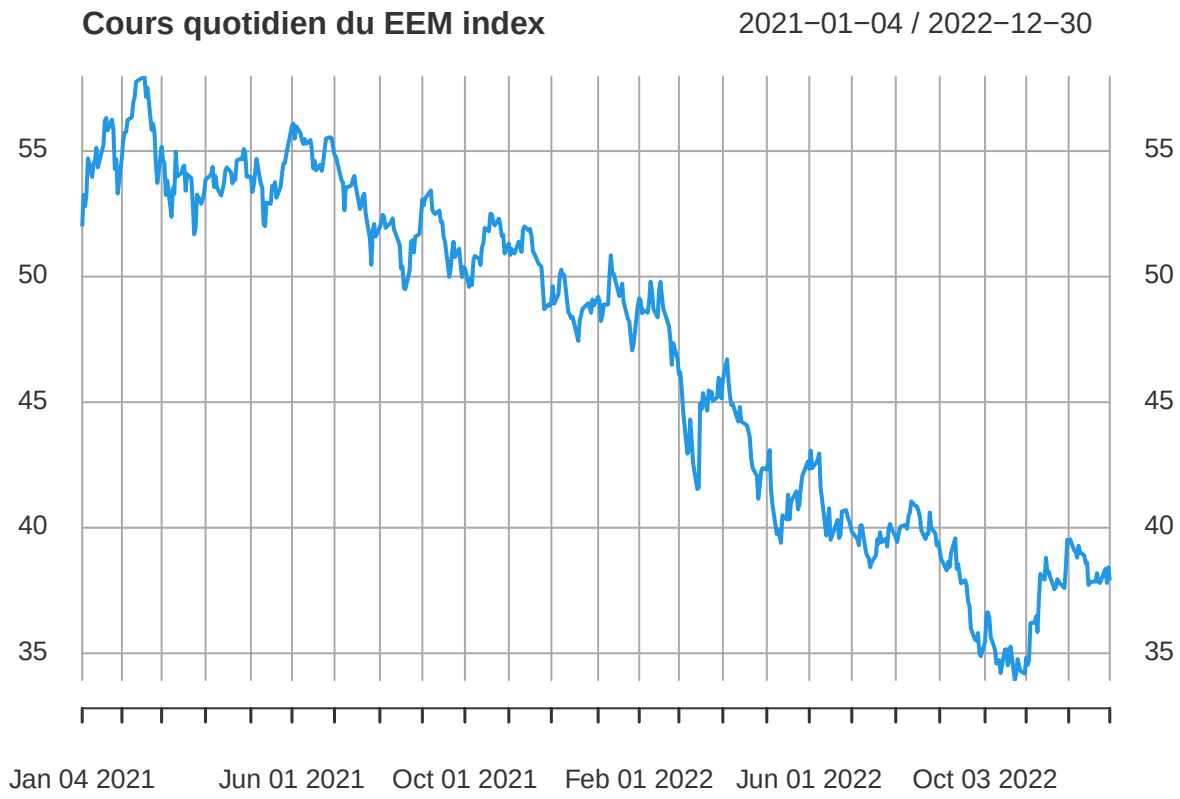
Les données que nous allons utiliser ont été téléchargées sur *yahoo.finance.com.*

```
library(quantmod)
library(TTR)
library(psych)
library(tseries)
library(knitr)
library(xtable)
library(lmtest)
library(forecast)
library(gridExtra)
library(ggplot2)
library(stats)
```

```
start_date <- "2021-01-01"
end_date <- "2022-12-31"
EEM <- getSymbols("EEM",
                  from = start_date,
                  to = end_date,
                  auto.assign = FALSE)
data <- Cl(EEM)
```

Cours quotidiens :

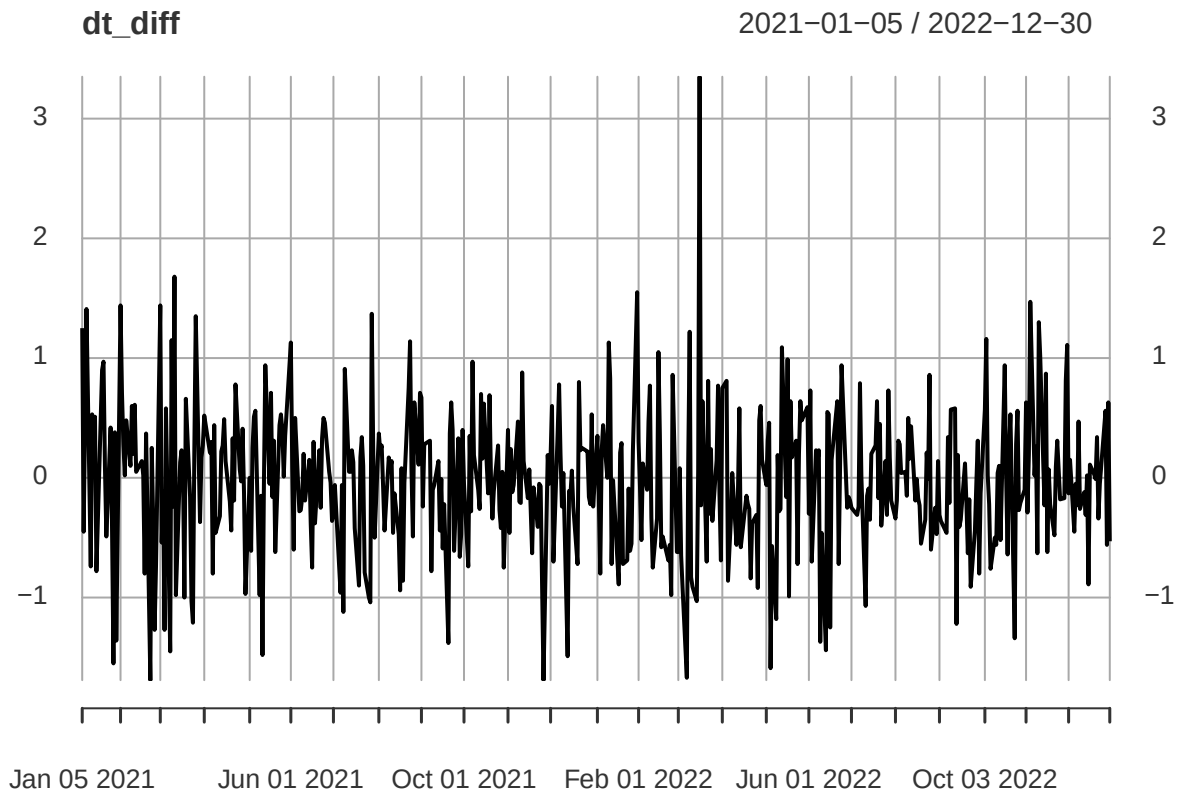
```
plot(data,
     main = "Cours quotidien du EEM index",
     type = "l",
     col = "4")
```



2. Stationnarité

```
dt_diff = diff(data)
dt_diff = na.omit(dt_diff)
adf1 = adf.test(dt_diff)
kpss1 = kpss.test(dt_diff)
```

```
plot(dt_diff)
```



Test	p.value	Stationnarité
ADF	0.01	Stationnaire
KPSS	0.10	Stationnaire

Le test ADF nous indique que la p-value (0.0001) est inférieure à 5%, et par conséquent, on doit rejeter l'hypothèse de la non stationnarité des données.

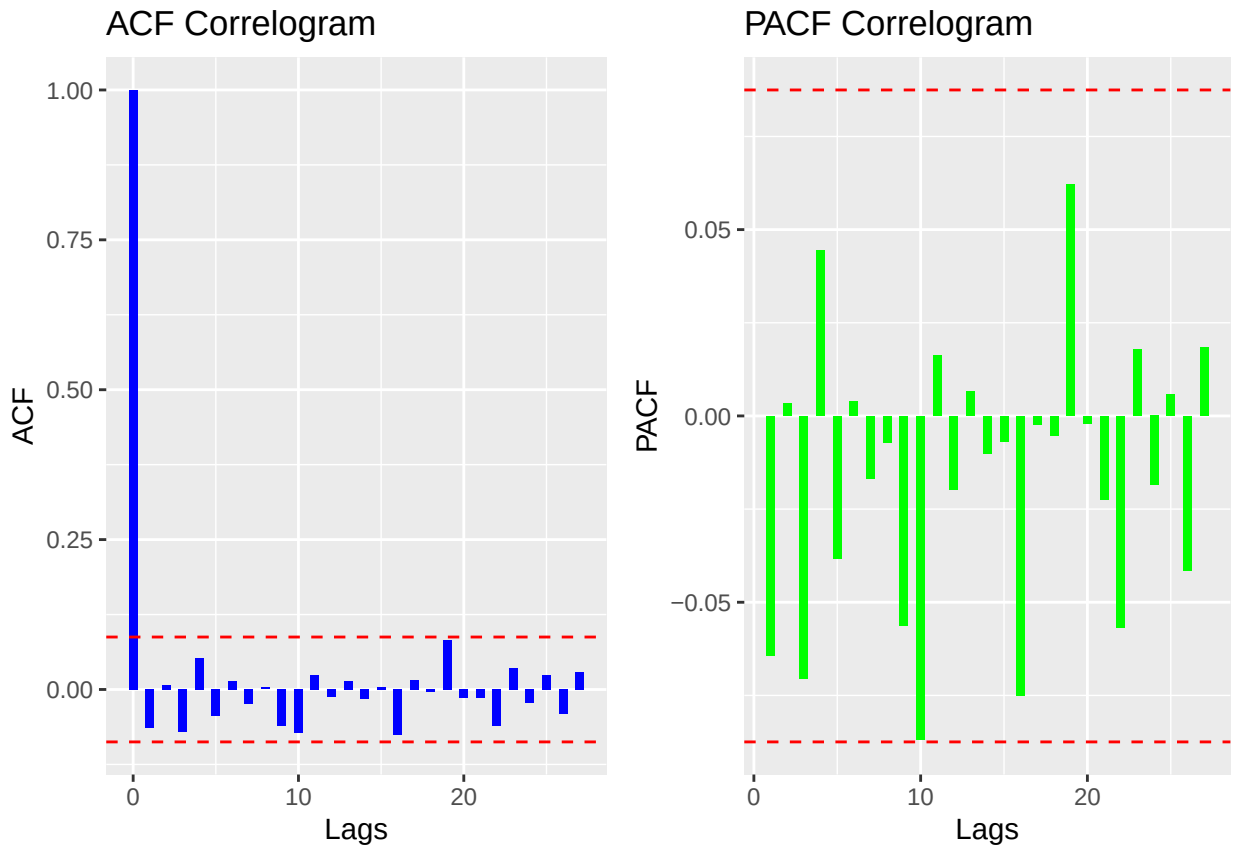
Le test KPSS nous indique aussi que les données sont stationnaires, puisque la p-value (0.1) est supérieure à 5% et donc on ne peut pas rejeter l'hypothèse de stationnarité des données.

Les deux tests montrent que les données des cours quotidiens sont stationnaires après la première différence $I(1)$.

3. Identification :

```
dt_diff <- as.data.frame(dt_diff)
# ACF
acf_data <- acf(dt_diff$EEM.Close, main = "ACF Correlogram", plot = FALSE)

# PACF
pacf_data <- pacf(dt_diff$EEM.Close, main = "PACF Correlogram", plot = FALSE)

# Création du graphique de l'ACF + bandes de significance
```

```r
acf_plot <- ggplot(data.frame(lag = acf_data$lag, acf = acf_data$acf), aes(x = lag, y = acf)) +
  geom_bar(stat = "identity", fill = "blue", width = 0.5) +
  geom_hline(yintercept = 1.96/sqrt(length(dt_diff$EEM.Close)), linetype = "dashed", color = "red") +
  geom_hline(yintercept = -1.96/sqrt(length(dt_diff$EEM.Close)), linetype = "dashed", color = "red") +
  labs(title = "ACF Correlogram") +
  xlab("Lags") +
  ylab("ACF")

# Création du graphique de PACF + bandes de significance
pacf_plot <- ggplot(data.frame(lag = pacf_data$lag, pacf = pacf_data$acf), aes(x = lag, y = pacf)) +
  geom_bar(stat = "identity", fill = "green", width = 0.5) +
  geom_hline(yintercept = 1.96/sqrt(length(dt_diff$EEM.Close)), linetype = "dashed", color = "red") +
  geom_hline(yintercept = -1.96/sqrt(length(dt_diff$EEM.Close)), linetype = "dashed", color = "red") +
  labs(title = "PACF Correlogram") +
  xlab("Lags") +
  ylab("PACF")

# Combiner les graphiques en un seul
combined_plot <- grid.arrange(acf_plot, pacf_plot, nrow = 1)
```



D'après les deux graphes, on peut conclure que le modèle ARMA approprié à cette série temporelle est un $ARIMA(1,1,1)$.

4. Estimation :

Nous allons utiliser dans ce travail la fonction *auto.arima*() du package $forecast$. cette fonction utilise utilise une variante de l'algorithme de *Hyndman-Khandakar* (Hyndman & Khandakar, 2008), qui combine des tests de racine unitaire, la minimisation de l'AICc et le MLE pour obtenir le modèle ARIMA le plus adéquat.

```
model_2 <- auto.arima(data,
                      seasonal = FALSE,
                      approximation = FALSE,
                      stepwise = TRUE,
                      nmodels = 5,
                      trace = TRUE,
                      method = c("CSS-ML")
                      )
```

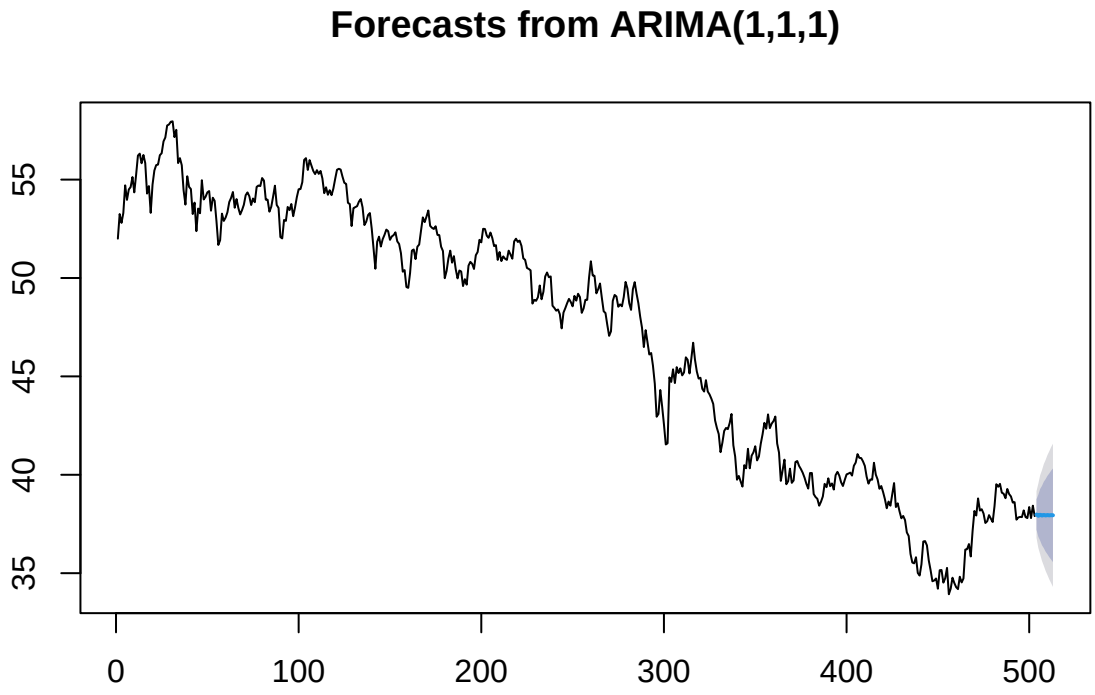
```
##
##  ARIMA(2,1,2) with drift         : 926.2772
##  ARIMA(0,1,0) with drift         : 927.5473
##  ARIMA(1,1,0) with drift         : 927.4702
##  ARIMA(0,1,1) with drift         : 927.4762
##  ARIMA(0,1,0)                    : 926.6047
##
##  Best model: ARIMA(2,1,2) with drift
```

Le meilleur modèle l'algorithme a choisi est un $ARIMA(2,1,2)$ avec non-zero mean (Drift).

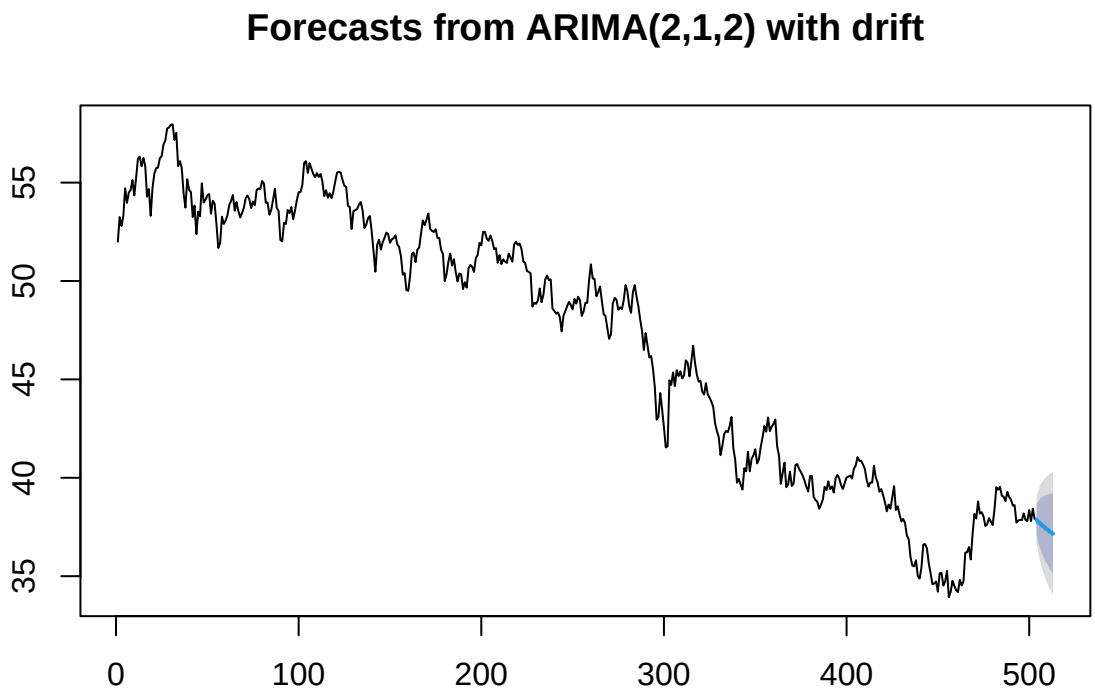
Nous allons retenir deux modèle à tester par la suite, le premier est un $ARIMA(2,1,2)$ avec une constante c , et le deuxième est un $ARIMA(1,1,1)$.

La moyenne (Drift) fait référence à un terme constant qui est ajouté à la série temporelle. Il représente la tendance à long terme ou le niveau général de la série chronologique.

```
model_1 <- arima(data,
                 order = c(1,1,1),
                 method = c("CSS-ML")
                 )
plot(forecast(model_1))
```




```
plot(forecast(model_2))
```



5. Diagnostics :

5.1. Normalité des résidus

```
norm_1 <- jarque.bera.test(resid(model_1))
norm_2 <- jarque.bera.test(resid(model_2))
```

```
norm_1
```

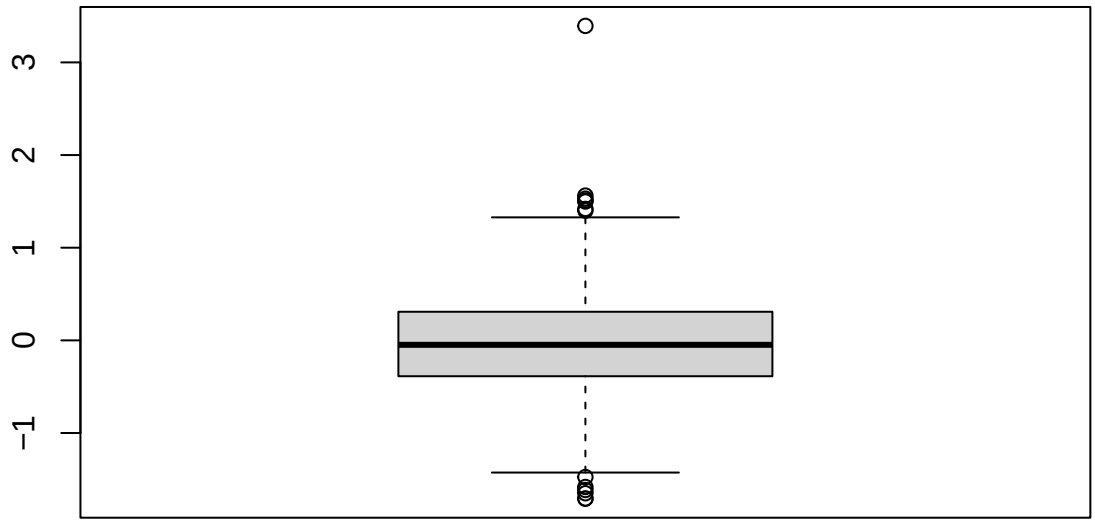
```
##
##  Jarque Bera Test
##
## data:  resid(model_1)
## X-squared = 82.294, df = 2, p-value < 2.2e-16
```

```
norm_2
```

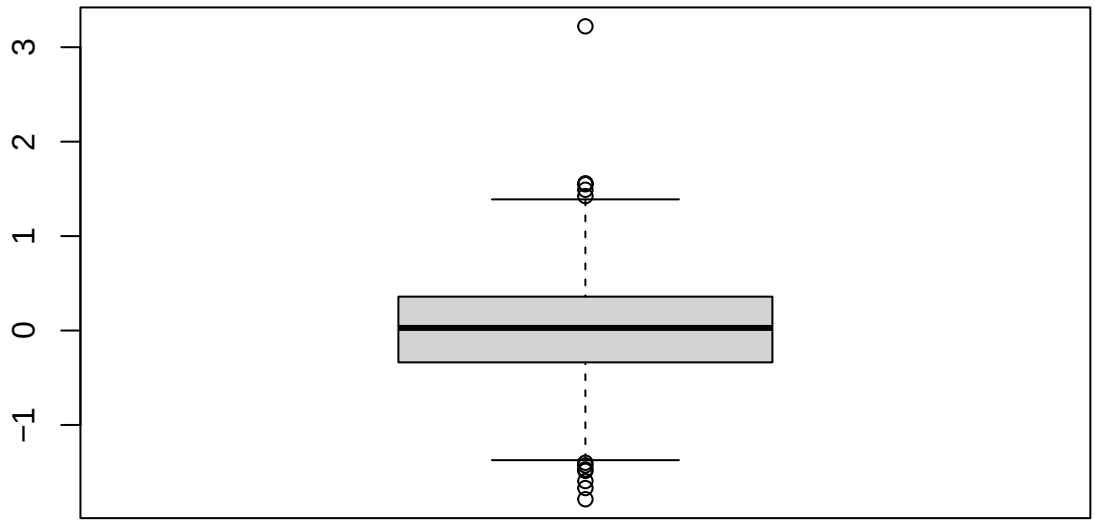
```
##
##  Jarque Bera Test
##
## data:  resid(model_2)
## X-squared = 50.395, df = 2, p-value = 1.14e-11
```

les p-value des tests de Jarque-Bera nous affirment que les résidus du deux modèles ne sont pas normalement distribués, ce qui signifie que que les deux modèles sont mal spécifiés ou qu'il y a des valeurs aberrantes dans les données. D'après le graphe, on voit clairement qu'il y a une seule valeur aberrante, ce qui peut être la cause de la non-normalité des résidus.

```
boxplot(resid(model_1))
```




```
boxplot(resid(model_2))
```



Le boxplot des résidus indiquent la présence des valeurs aberrantes.

5.2. Autocorrélation des résidus

```
residuals_1 = checkresiduals(model_1, plot = FALSE)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 7.3064, df = 8, p-value = 0.504
##
## Model df: 2.   Total lags used: 10
```

```
residuals_2 = checkresiduals(model_2, plot = FALSE)
```

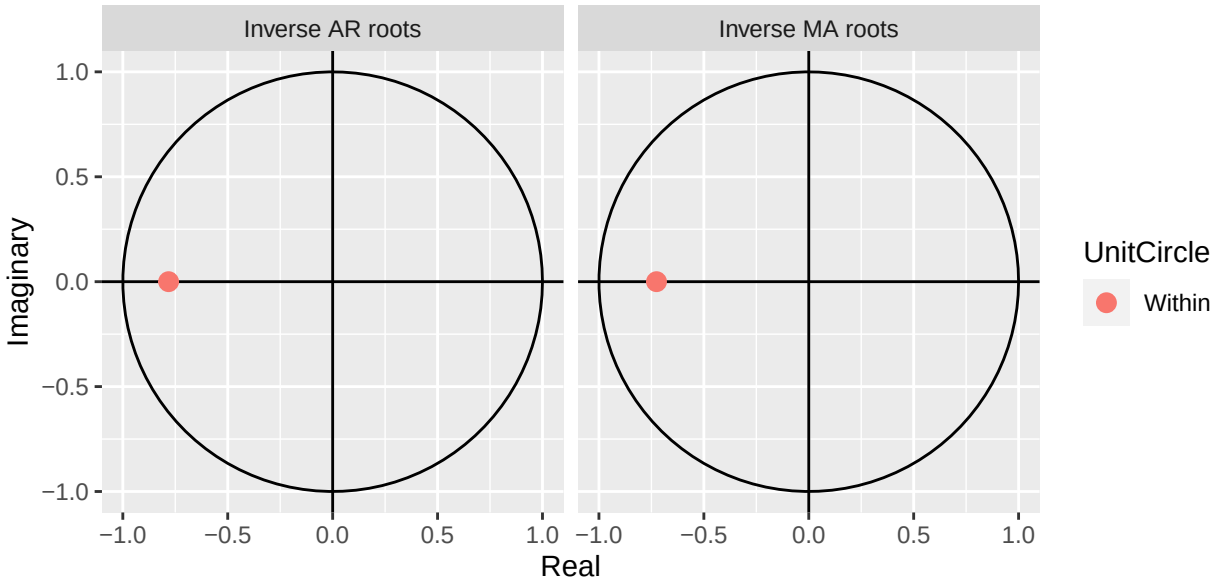
```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,2) with drift
## Q* = 4.5997, df = 6, p-value = 0.5961
##
## Model df: 4.   Total lags used: 10
```

Le test de Ljung-box indique qu'il n'y a pas de preuve d'autocorrélation significative dans les résidus des deux modèles de série temporelle évalués. En d'autres termes, l'hypothèse nulle selon laquelle les résidus ne sont pas corrélés jusqu'au décalage spécifié n'est pas rejetée.

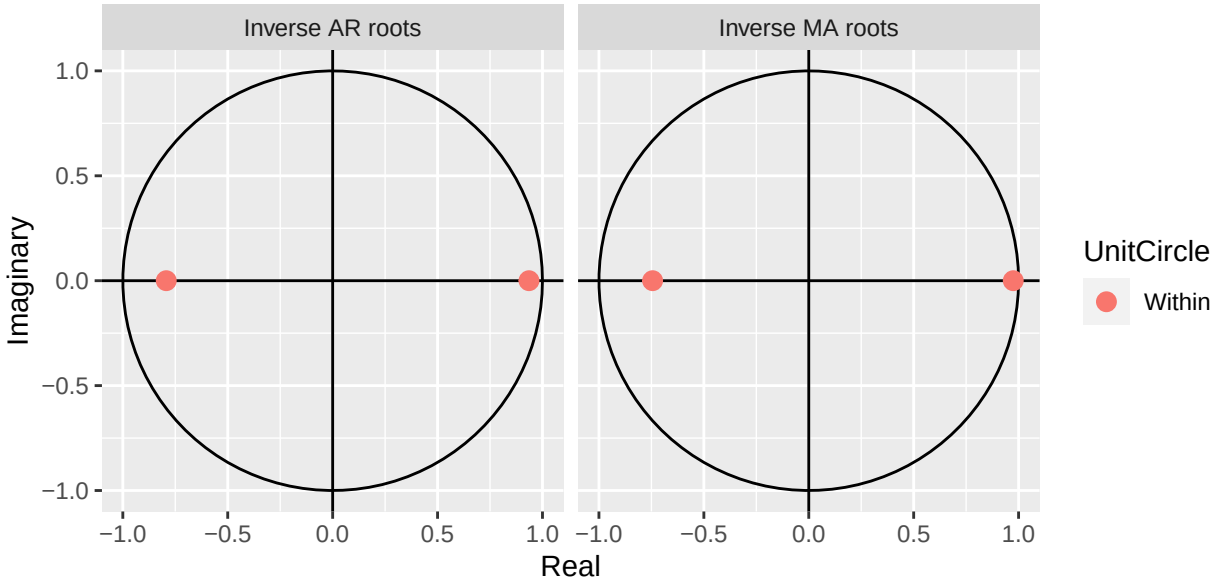
Ceci est généralement considéré comme une bonne nouvelle, car cela suggère que les deux modèles capturent de manière adéquate la dépendance temporelle des données et que les résidus ne présentent pas d'autocorrélation significative. Cependant, il est important de noter que l'absence d'autocorrélation significative n'implique pas nécessairement que le modèle est le meilleur ajustement possible pour les données, et d'autres tests de diagnostic doivent être utilisés pour évaluer l'adéquation globale du modèle.

5.3. Invertibilité :

```
layout(1:3)
autoplot(model_1)
```




```
autoplot(model_2)
```



Les inverses des racines AR et MA de l'équation caractéristique peuvent être utilisées pour vérifier si le processus impliqué par le modèle est stationnaire et inversible. Pour que les parties AR et MA du processus soient stationnaires et inversables, respectivement, les racines inversées doivent dans chaque cas être inférieures à 1 en valeur absolue, ce qui est le cas ici. un en valeur absolue, ce qui est le cas ici.

5.4. Stationarité des résidus

```
resid_adf_1 = adf.test(resid(model_1))
resid_adf_2 = adf.test(resid(model_2))
resid_kpss_1 = kpss.test(resid(model_1))
resid_kpss_2 = kpss.test(resid(model_2))
```

model	Test.ADF	Test.KPSS	Stationnarité
ARIMA(1,1,1)	0.01	0.1	Stationnaire
ARIMA(2,1,2)	0.01	0.1	Stationnaire

La p-value du test ADF est inférieure à 5% (0.0001 pour les deux modèles), ce qui signifie l'absense d'une racine unitaire, et donc les résidus sont stationnaires.

C'est le même cas pour le test KPSS; la p-value est supérieure à 5%, ce qui implique qu'on peut rejeter l'hypothèse null de la non-stationnarité des résidus, et ce pour les deux modèles pris en considération.

5.5. Evaluation des prévisions :

```
kable(accuracy(forecast(model_1)))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.0287511	0.6046785	0.4611094	-0.0734627	0.9983254	0.9907418	-0.0019322

```
kable(accuracy(forecast(model_2)))
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.0084297	0.6006804	0.4589852	-0.0023872	0.9925201	0.9861777	0.0179708

Les résultats obtenus montrent que le premier modèle $ARIMA(2,1,2)$ minimise $RMSE$; MAE; MPE ; $MAPE$; et le $MASE$.

Ce qui signifie que le modèle $ARIMA(2,1,2)$ que l'algorithme a choisi et le plus adéquat.

5.6. Heteroscedasticité:

```
library(aTSA)
hetero_1 = arch.test(model_2, output = FALSE)
hetero_2 = arch.test(model_1, output = FALSE)

hetero_1
```

```
##      order       PQ     p.value        LM      p.value
## [1,]     4 10.25422 0.036356284 292.11221 0.000000e+00
## [2,]     8 23.77921 0.002495442 133.28731 0.000000e+00
## [3,]    12 25.93130 0.010976683  85.81039 1.098011e-13
## [4,]    16 30.54008 0.015395069  61.76970 1.248577e-07
## [5,]    20 31.58425 0.047934847  49.54551 1.528351e-04
## [6,]    24 33.31567 0.097586044  40.70235 1.280472e-02
```

```
hetero_2
```

```
##      order        PQ    p.value        LM      p.value
## [1,]     4  8.232533 0.08342212 324.97294 0.000000e+00
## [2,]     8 19.798543 0.01112570 149.57577 0.000000e+00
## [3,]    12 21.791984 0.03991840  96.30032 9.992007e-16
## [4,]    16 26.592106 0.04624556  68.66913 7.702112e-09
## [5,]    20 27.848113 0.11304390  55.25504 2.124301e-05
## [6,]    24 29.559894 0.19976897  45.21516 3.740171e-03
```

D'après le test de Langrange-multiplier et de Portmanteau-Q des deux modèles, on voit bien que les p-values sont inférieures à 5%. Cela suggère que les modèles sont inadéquats et qu'ils existent toujours un modèle ou une structure significative dans les résidus qui n'a pas été capturé par le modèle.

Ce résultat implique que les modèles doivent être révisés ou améliorés pour mieux saisir les modèles sous-jacents dans les données.

Il peut aussi suggérer que les données elles-mêmes sont plus complexes qu'on ne l'avait supposé à l'origine et qu'il faut un modèle plus complexe ou plus sophistiqué pour saisir les modèles sous-jacents. Une analyse et une exploration plus approfondies des données peuvent être nécessaires pour déterminer la spécification appropriée du modèle et les techniques d'estimation des paramètres.

Réferences :

[1] Box, George E. P., Gwilym M. Jenkins, and Gregory C. Reinsel. Time Series Analysis: Forecasting and Control. 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1994.

[2] Brooks, C. (2012) Introductory Econometrics for Finance. 3rd edn. Cambridge University Press.