Compte-rendu de TP 1 – Modélisation de séries temporelles

Émilie CAILLERIE
Véronique DEMIANENKO

```r
rm(list=ls())

library(quantmod)
library(tseries)
library(forecast)
library(zoo)
```

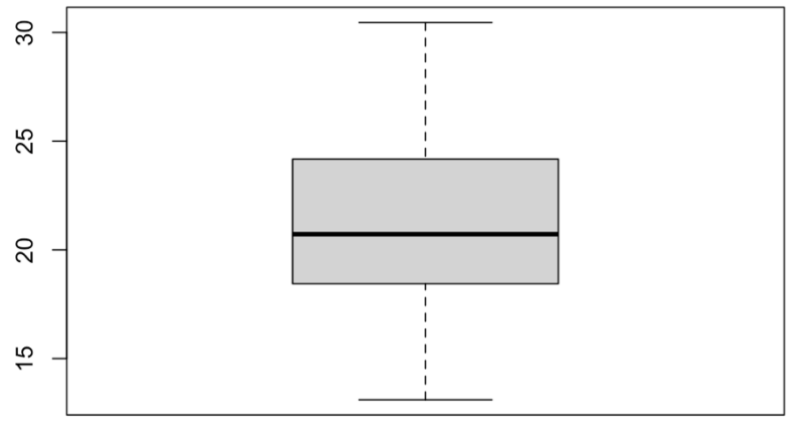
1) Analyse descriptive

```r
#1)
debut <- as.Date("2020-01-01")
fin <- as.Date("2022-12-31")

getSymbols("GOLD", from = debut, to = fin)
View(GOLD)


#2)
plot(GOLD$GOLD.Close, col = "pink", xlab = "Date", ylab = "Prix de clôture de l'or")

# Analyse descriptive :
mean(GOLD$GOLD.Close,na.rm= TRUE)
sd(GOLD$GOLD.Close,na.rm= TRUE)
summary(GOLD$GOLD.Close)
boxplot(GOLD$GOLD.Close)
```

Voici le prix de l’or en temps réel du 01/01/2020 au 31/12/2022 :

On obtient ici la moyenne, les quartiles, la médiane, le min et le max ainsi que la boîte à moustache :

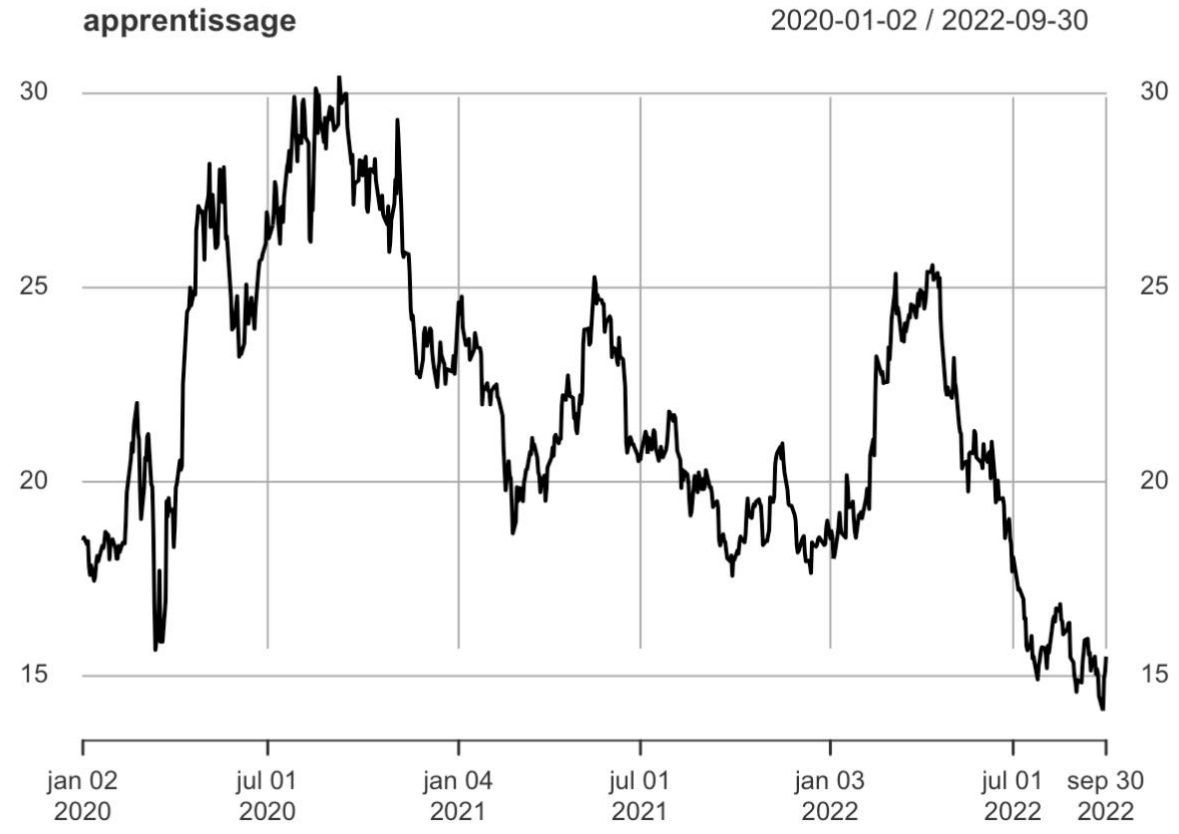
```
> # Analyse descriptive :
> mean(GOLD$GOLD.Close,na.rm= TRUE)
[1] 21.30594
> sd(GOLD$GOLD.Close,na.rm= TRUE)
[1] 3.967074
> summary(GOLD$GOLD.Close)
     Index              GOLD.Close
 Min.   :2020-01-02   Min.   :13.10
 1st Qu.:2020-09-30   1st Qu.:18.44
 Median :2021-07-01   Median :20.72
 Mean   :2021-07-01   Mean   :21.31
 3rd Qu.:2022-03-31   3rd Qu.:24.17
 Max.   :2022-12-30   Max.   :30.46
> boxplot(GOLD$GOLD.Close)
```



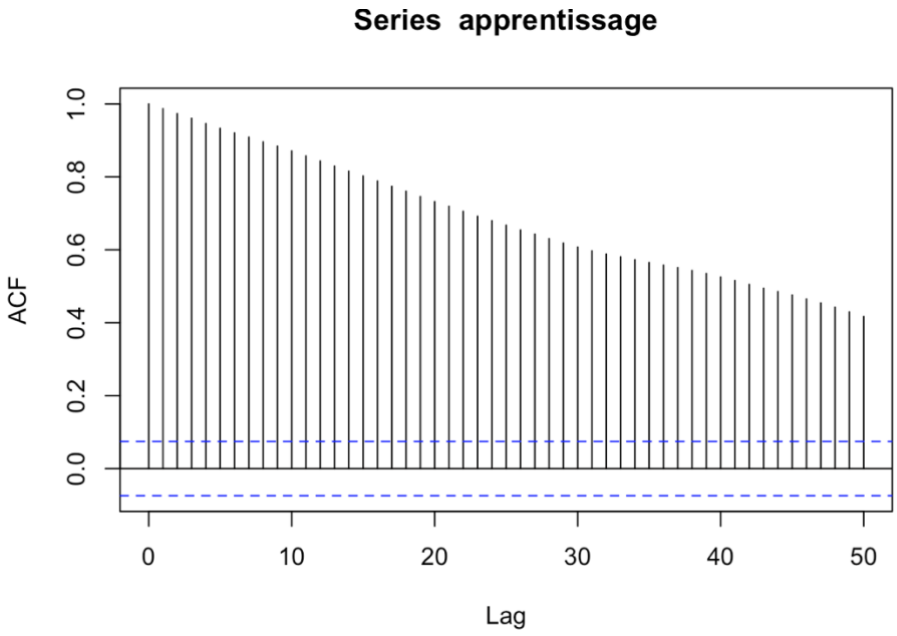
2) Modélisation

```
29
30  # On divise les données en deux parties
31  fin2 <- as.Date("2022-10-01")
32  getSymbols("GOLD", from = debut, to = fin2)
33
34  apprentissage <- na.omit(GOLD$GOLD.Close)
35  plot(apprentissage)
36
37  #1a)
38  #ACF
39  acf_result <- acf(apprentissage, lag.max = 50)
41  #PACF
42  pacf_result <- pacf(apprentissage, lag.max = 50)
```

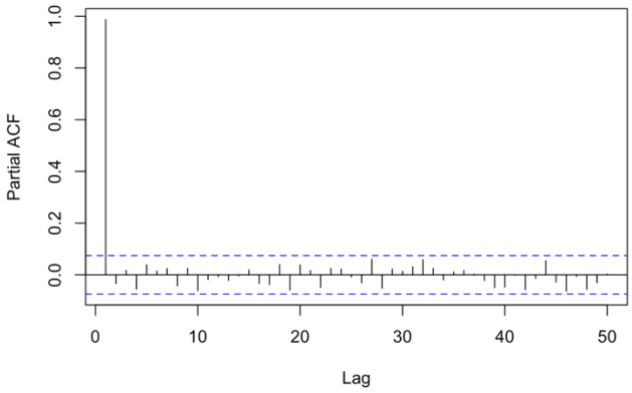
Voici le tracé du prix de l'or jusqu'au 1/10/2022 :



En traçant l'ACF, on observe une décroissance régulière, ce qui indique une dépendance à court terme de la série chronologique.



En traçant le PACF, on observe l'existence d'un seul pic significatif, suivi de corrélations non significatives. Cela indique donc un terme AR d'ordre 1.

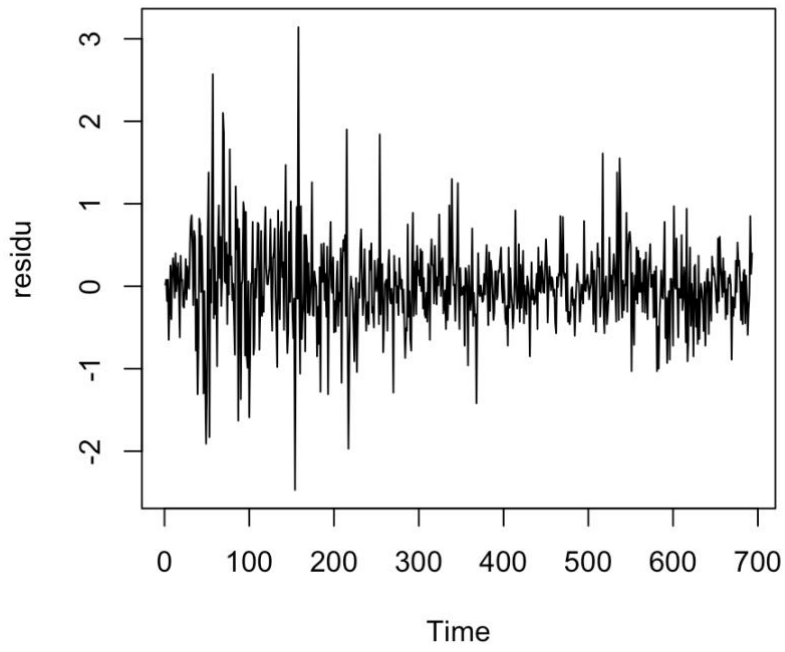



```
#1b)
model = auto.arima(apprentissage)
model
```

Le modèle ARIMA proposé par la machine est ARIMA(0,1,0).

On extrait ensuite la série résiduelle :

```
#1c)
# Extraction de la série résiduelle
residu <- residuals(model)
plot(residu)

```



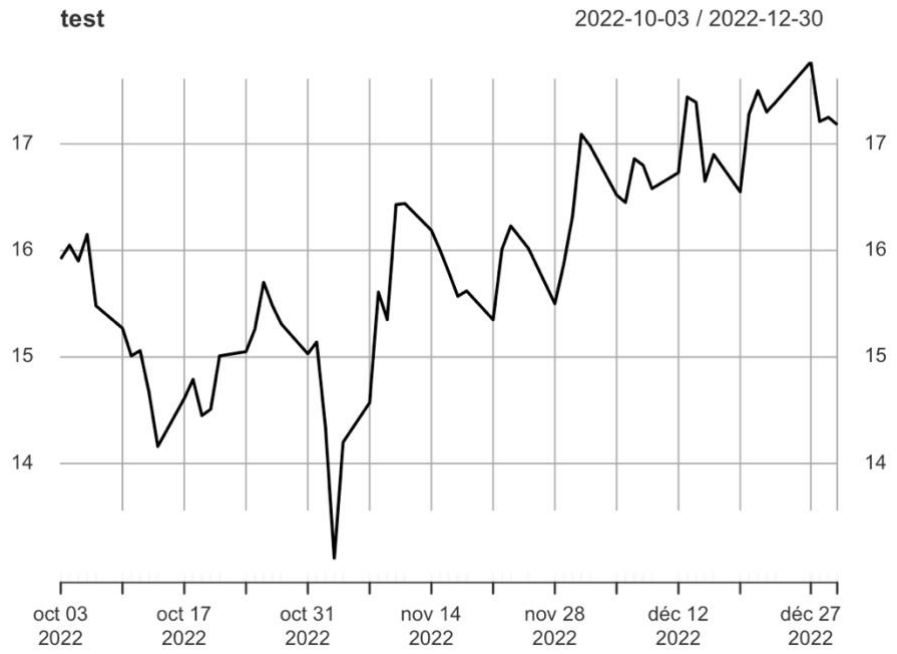
```r
#1c)
# Extraction de la série résiduelle
residu <- residuals(model)
plot(residu)

# Test d'indépendance
ljung_box_test <- Box.test(residu, lag = 5, type = "Ljung-Box")
ljung_box_test
ljung_box_test$method


# Test de stationnarité
adf_test <- adf.test(residu)
adf_test

```

En réalisant le test d'indépendance, la p-valeur vaut 0.3918, ce qui est relativement élevé. Cela signifie que l'autocorrélation n'est pas significative. Cela montre l'indépendance des termes les uns par rapport aux autres.

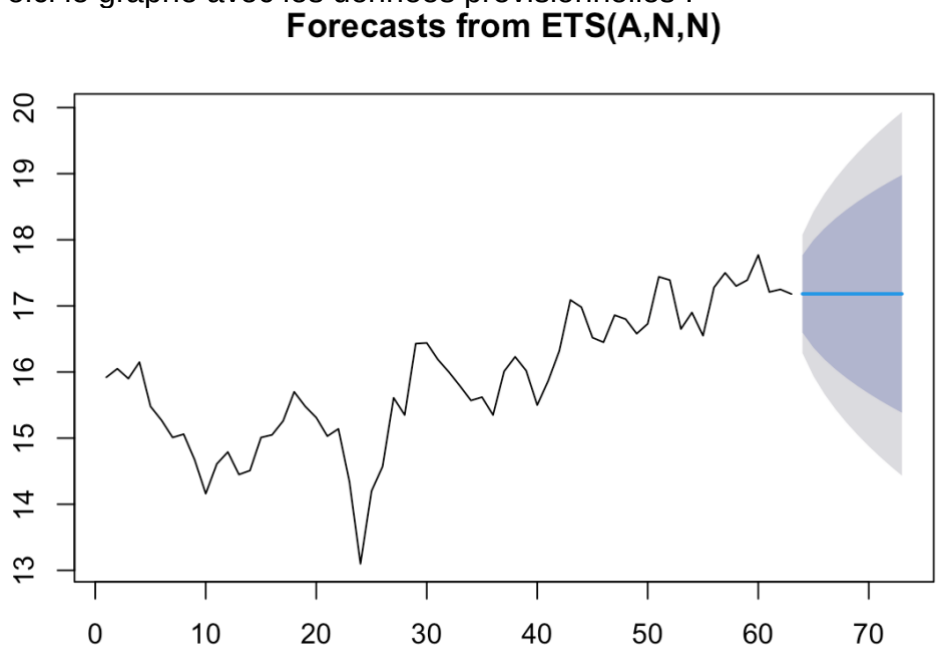
En réalisant le test de stationnarité, la p-valeur vaut 0.01, ce qui est faible. La série résiduelle est donc stationnaire, ce qui est confirmé par le test lui-même.

```r
#2)

getSymbols("GOLD", from = fin2, to = fin)

test <- na.omit(GOLD$GOLD.Close)
plot(test)

# On test le modèle ARIMA trouvé précédemment sur l'ensemble de données test
# (de octobre à décembre)
arima(test, c(0,1,0))

forecast(test)
plot(forecast(test))
forecast <- na.omit(forecast)

```

Voici le graphe avec les vraies données :



Voici le graphe avec les données prévisionnelles :



On observe que l'allure des deux graphes est identique, le modèle ARIMA(0,1,0) est donc bien adapté.

```
86  # Calcul de l'erreur quadratique moyenne
87
88  eqm <- sqrt(mean((forecast(test)$mean - residu)^2))
89  eqm
90
91  # Calcul de l'erreur absolue moyenne
92  eam <- mean(abs(forecast(test)$mean - residu))
93  eam
```

Finalement, on calcule l'erreur quadratique moyenne et l'erreur absolue moyenne, et on obtient respectivement les valeurs suivantes :

```
> # Calcul de l'erreur quadratique moyenne
>
> eqm <- sqrt(mean((forecast(test)$mean - residu)^2))
> eqm
[1] 16.58667
>
> # Calcul de l'erreur absolue moyenne
> eam <- mean(abs(forecast(test)$mean - residu))
> eam
[1] 16.56919
```