Projet STA202

Mohamed Issa et Mehdi Ghrabli

28/02/2021

Objectifs

L'objectif de ce TP est d'appliquer les méthodes vues dans le cours STA202 sur un jeu de données. Le jeu de données choisi dans notre TP est le cours boursier de l'action de l'entreprise "apple" téléchargé à partir du site yahoo finance (https://fr.finance.yahoo.com/quote/AAPL/history?p=AAPL).

Description du jeu de données

Le site offre des données quotidiennes à partir du 12 décembre 1980 jusqu'à présent mais on a choisi de prendre seulement les données des 4 dérniéres années, c'est à dire du 28 février 2017 jusqu'au 28 février 2021. Les données disponibles sont :

1. La date
2. le cours d'ouverture qui est le cours à l'ouverture de la bourse
3. le cours de clotûre qui est le cours à la fermeture de la bourse
4. le cours elevé et faible qui sont respectivement le prix maximal et le prix minimal atteints par l'action
5. le cours de clotûre ajusté
6. le volume qui est la valeur totale des transactions des actions apple.

Dans ce projet, on va s'interesser seulement au cours de clotûre vu que c'est la donnée la plus utilisée pour determiner la performance durant la période. Elle est aussi la donnée de réference à n'importe quelle date choisie.

Chargement des données

```
setwd("D:")
data=read.csv("apple.csv")
```

Mise en forme des données

création de la séquence des dates

```
#Les variables dates:
df=data$Close
n=length(df)
date=c(1:n)
date1<- strptime("02/28/2017", "%m/%d/%Y")
date2<- strptime("02/28/2021", "%m/%d/%Y")
allDates<-seq(date1,date2, by = "day")

allValues  <- merge(
  x=data.frame(Date=allDates),
  y=df,
  all.x=TRUE)
```

En comparant la longueur de la séquence des dates et celles des données, on remarque qu'on a des données manquantes à certaines dates.

```
len_dates=length(allDates)
len_Close=length(df)
```

On doit donc procéder à un traitement de données pour ajouter des valeurs adéquates aux dates qui manquent d'informations.

Repérer les valeurs manquantes par NA

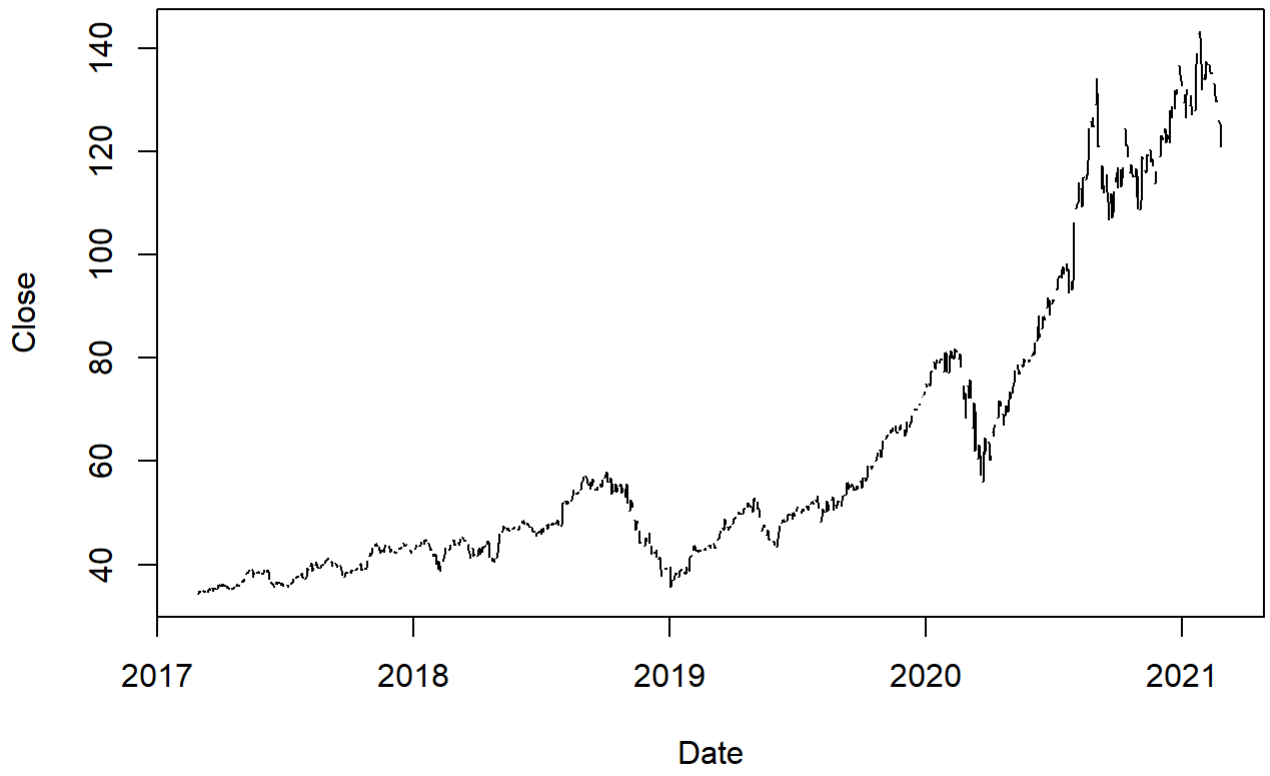
```
Dates_format=format(allDates, "%Y-%m-%d")
Valeurs=data$Close
Dates=data$Date
Cl=rep(0,len_dates)
compt=c(1:len_dates)
for (i in compt){
  if (Dates_format[i] %in% Dates){
    Cl[i]=Valeurs[1]
    Valeurs=Valeurs[-1]
  }
  else{
    Cl[i]=NA
  }
}
```

création de la data frame

```
data=data.frame(Date=allDates,Close=Cl)
```

plot de la série avant traitement

```
plot(data,type='l')
```



Pour ajouter des valeurs aux dates où on ne dispose pas d'information, on utilise la fonction na_interpolation du package imputeTS. Elle ajoute des valeurs par interpolation aux endroits où il y a NA.

Correction de la data frame

```
data=na_interpolation(data)
```

Série temporelle après traitement

```
d.xts=xts(data$Close,order.by=allDates)
```

plot de la série après traitement

Analyse descriptive

Une premiére étape est d'évaluer les valeurs pertinents dans notre jeu de donnée notamment la moyenne μ, l'écart type σ et le rapport $\frac{\sigma}{\mu}$. En fait ce rapport est appelé le coefficient de variation qui nous donne une idée sur la dispersion des données (on divise par μ pour avoir une echelle du problème)

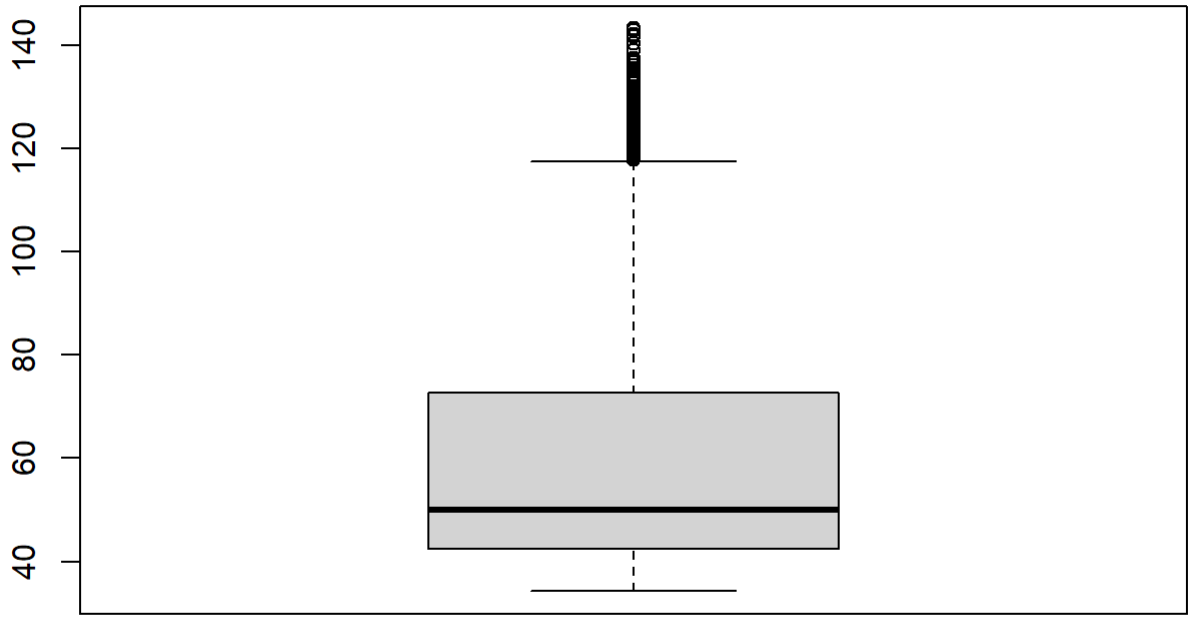
```
m=mean(data$Close)
sd=sd(data$Close)
sd/m
```

```
## [1] 62.04711
```

```
## [1] 28.51018
```

```
## [1] 0.4594925
```

On observe que la valeur du coefficient de variation est très élevée (loin de 0) donc les données sont très éloignées de la valeur moyenne. Une autre façon plus graphique de voir la disposition des données est de tracer la boîte à moustaches:



Boite à moustache des valeurs de clotûre

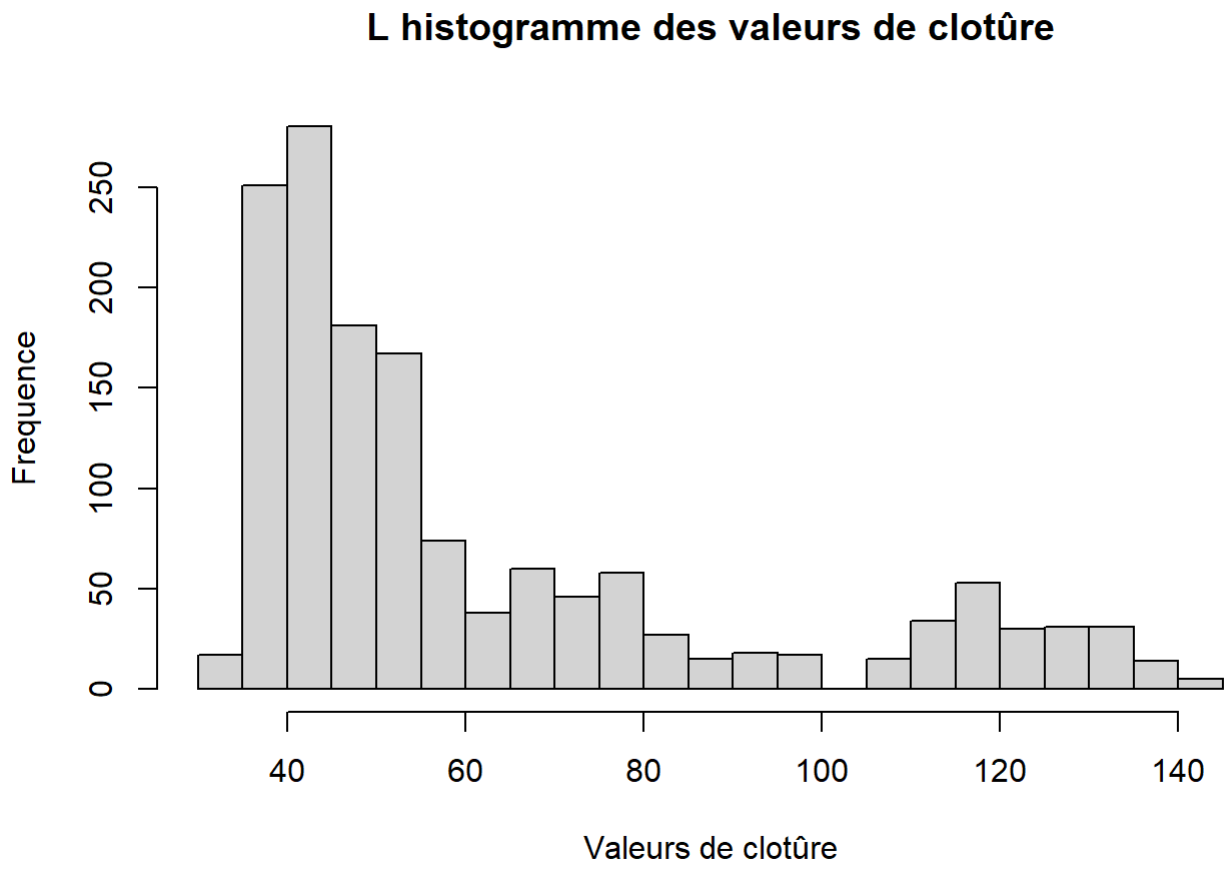
La boîte à moustaches indique une valeur moyenne aux alentours de 50 qui est inférieure à celle calculée precedemment (μ), c'est dû au fait que la boîte ignore les valeurs distantes appelées "outliers". Ces valeurs correspondent au pics atteints par la valeur des actions de apple au début de séptembre 2020 et fin janvier 2021. pour voir tout les détails de la boîte, on utilise l'instruction suivante où v est la variable qui contient la boîte elle même.

```
v$stats
```

qui indique respectivement : La valeur minimale, la premiére quantile, la valeur moyenne, la troisiéme quantile et la valeur maximale:

```
##           [,1]
## [1,]  34.24750
## [2,]  42.52333
## [3,]  50.05500
## [4,]  72.60833
## [5,] 117.51000
```

On peut aussi visualiser les données à l'aide d'un histogramme, on fixe ici la pas égale à 20 :



Les valeurs de clotûre sont concentrées entre 40 et 60 ce qui explique la forme de la boîte à moustaches.

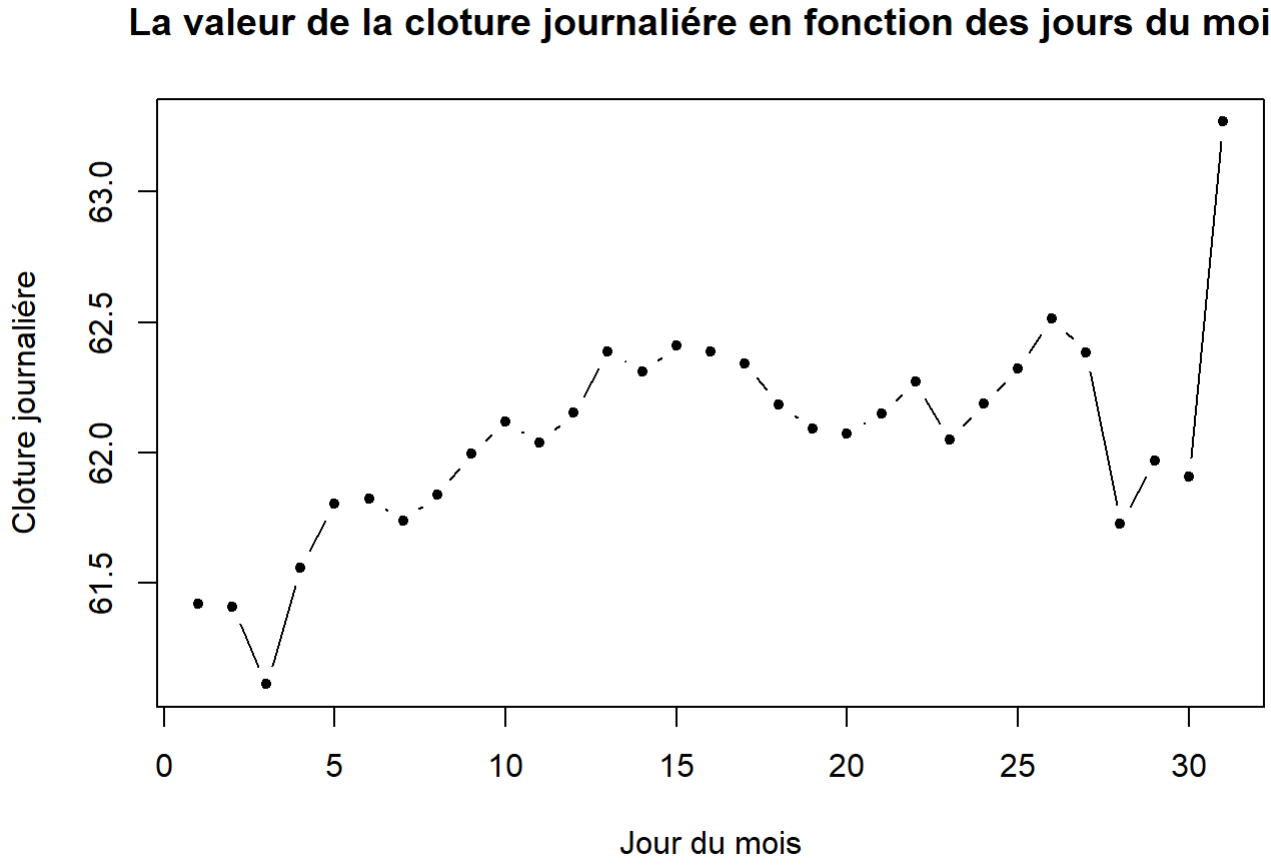
Analyse selon les périodes

On peut effectuer une analyse des données selon les périodes afin d'estimer une éventuelle saisonalité.

Analyse selon les jours

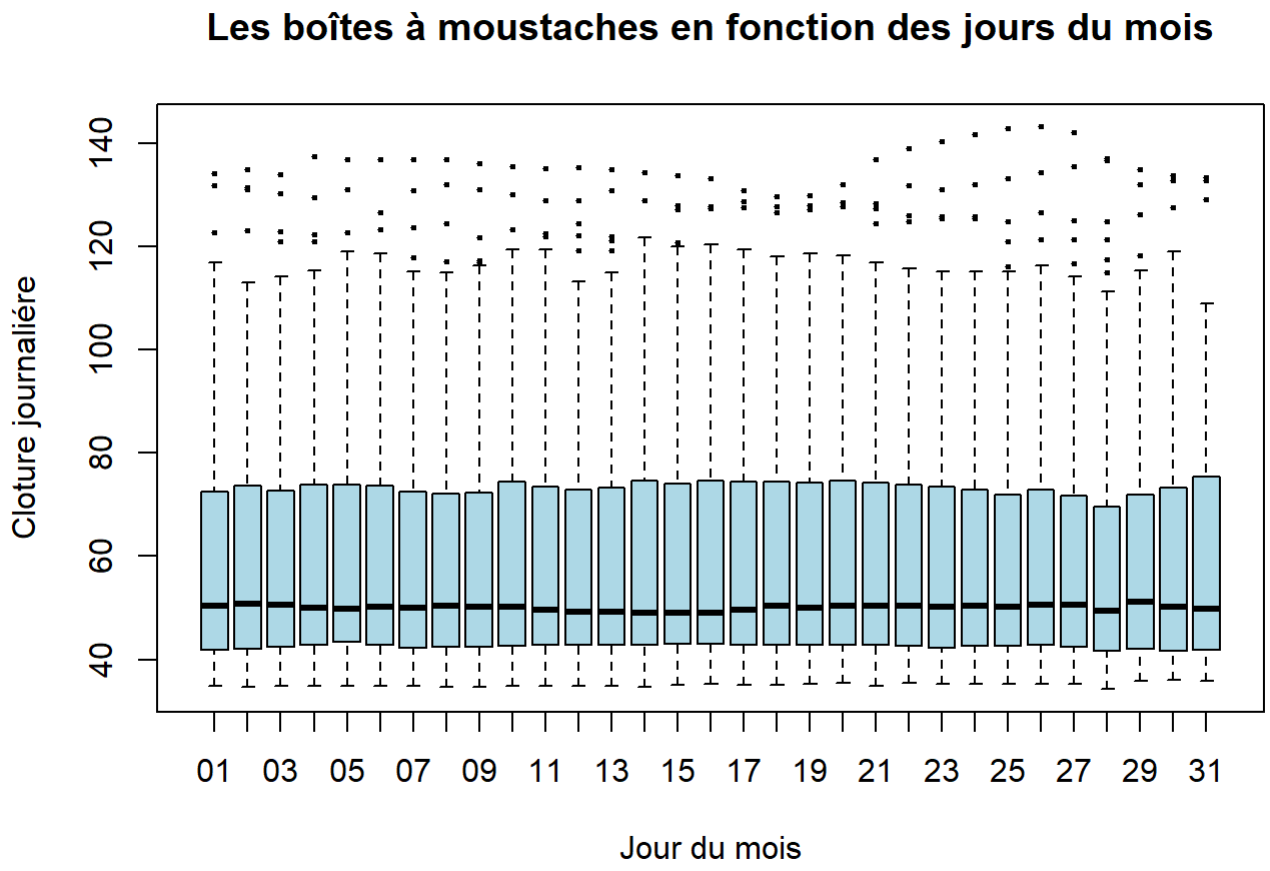
On trace d'abord la courbe de la moyenne de la valeur de clotûre selon les jours du mois:

```
jour <- as.factor(format(data$Date, "%d"))
ClotûreJournaliére <- tapply(data$Close, jour, mean)
plot(c(1:31),ClotûreJournaliére, type = "b", pch = 20,xlab="Jour du mois",main="La valeur de la
 cloture journaliére en fonction des jours du mois",ylab="Cloture journaliére")
```



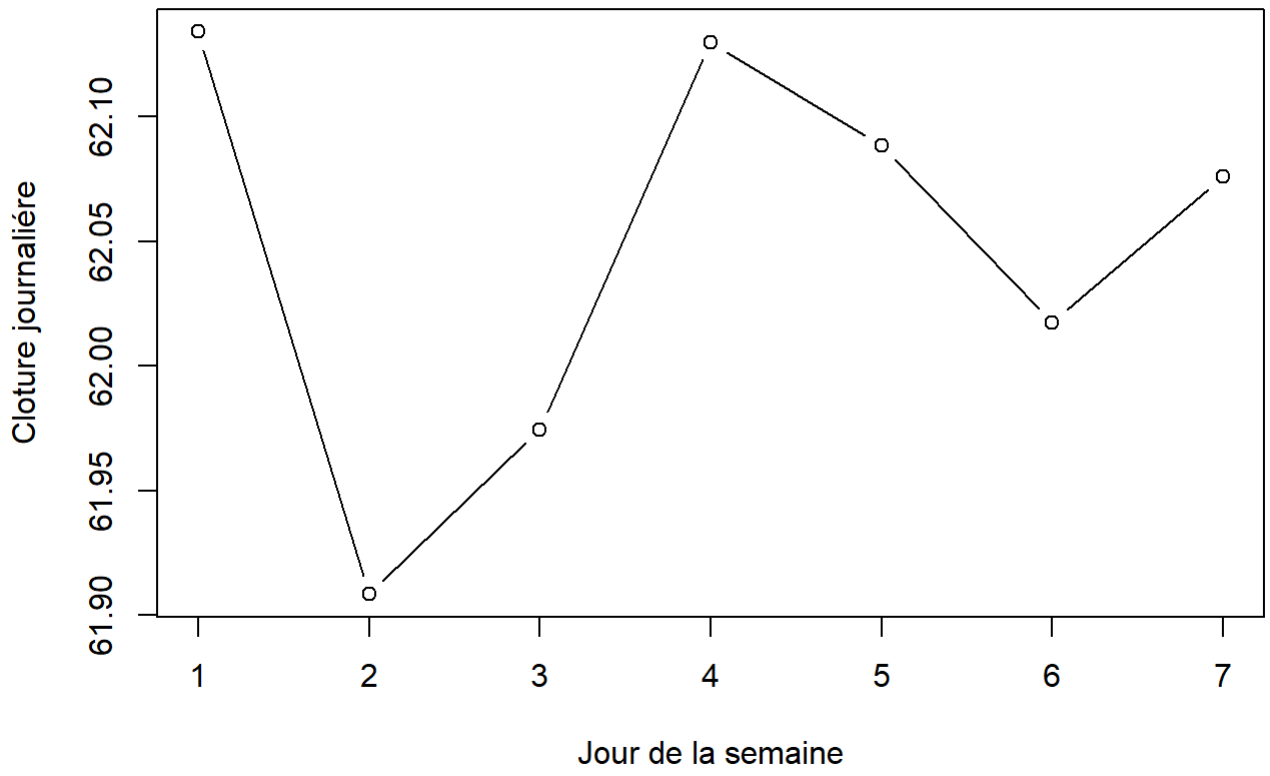
On observe que la valeur moyenne est maximale vers la fin du mois et minimale vers le début, mais les valeurs sont très proches de la moyenne pour pouvoir tirer une conclusion. On peut aussi tracer le boxplot pour chaque jour du mois :

```
boxplot(data$Close ~ jour, col = "lightblue", pch = 20, cex = 0.5,xlab="Jour du mois",ylab="Clot
ure journaliére",main="Les boîtes à moustaches en fonction des jours du mois")
```

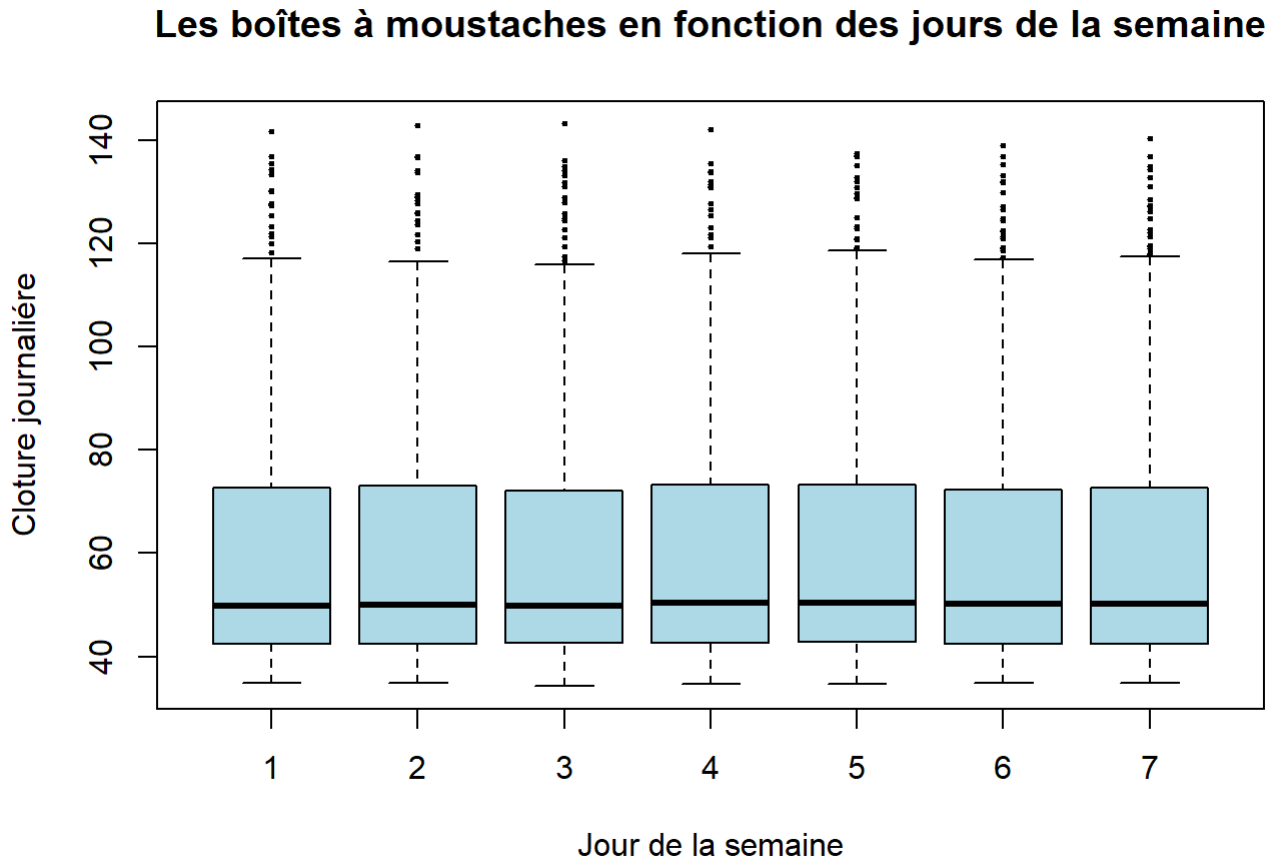


On peut effectuer le même travail mais sur les jours de la semaine pour pouvoir determiner une sorte d'activité au cours de la semaine.

```
dow<-as.factor(.indexwday(d.xts)+1)
Close_day<-tapply(d.xts,dow,mean)
plot(Close_day,type='b',xlab="Jour de la semaine",ylab="Cloture journaliére",main="La valeur de
 la cloture journaliére en fonction des jours de la semaine")
```



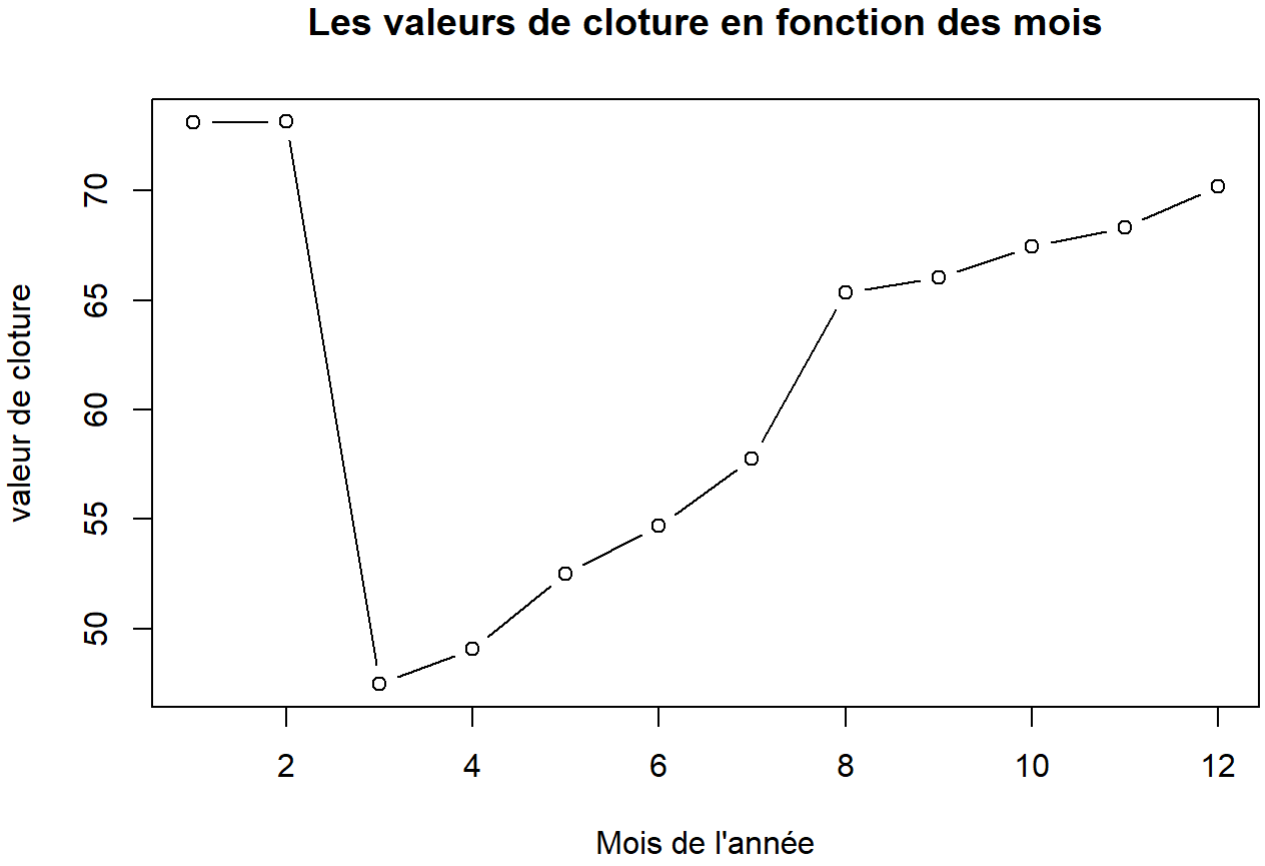

```
boxplot(data$Close ~ dow, col = "lightblue", pch = 20, cex = 0.5,xlab="Jour de la semaine",ylab=
"Cloture journaliére",main="Les boîtes à moustaches en fonction des jours de la semaine")
```



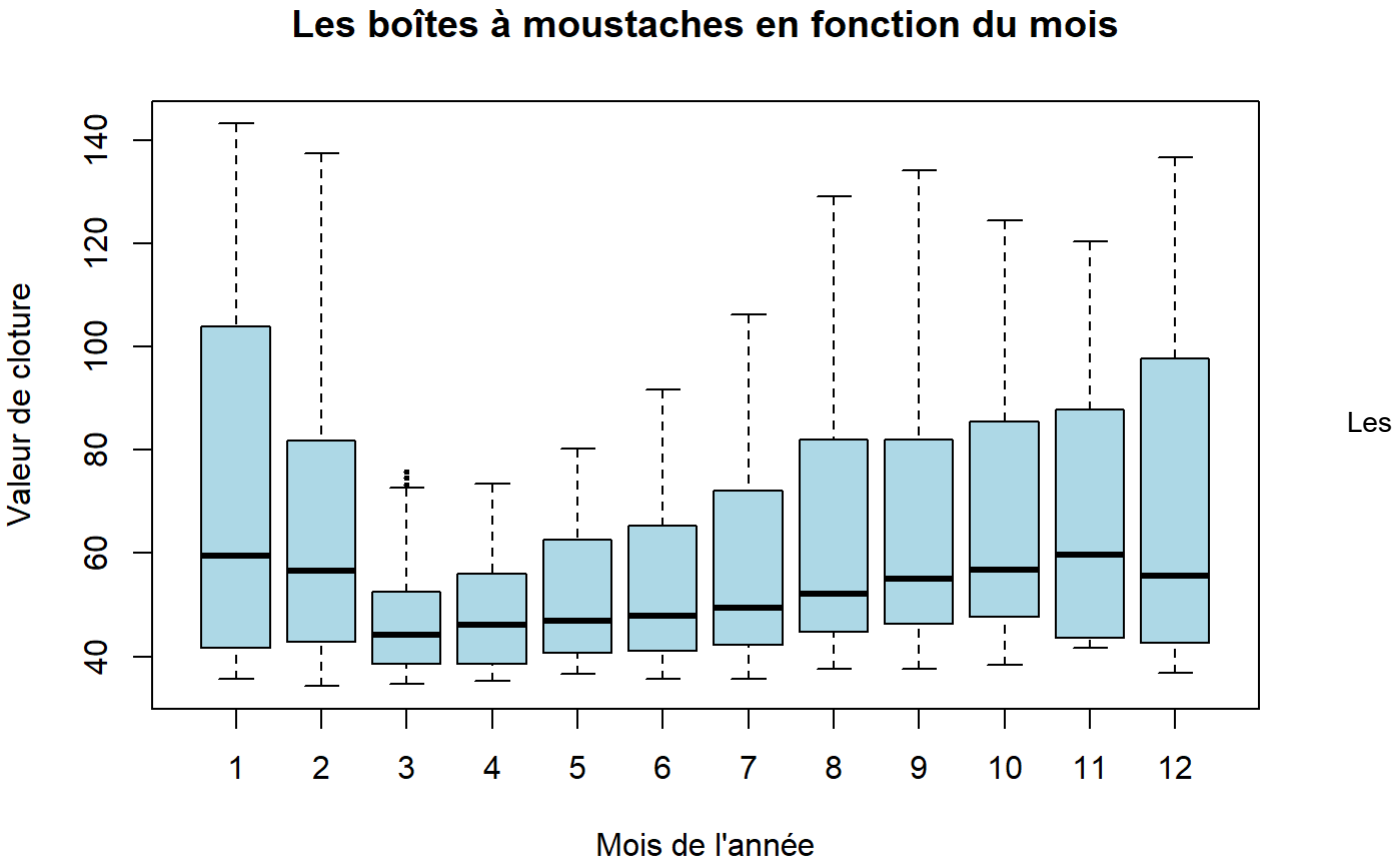
Les valeurs sont encore trop serrées pour pouvoir créer une hypothése de saisonalité.

Analyse selon les mois

```
mo<-as.factor(.indexmon(d.xts)+1)
Close_mon<-tapply(d.xts,mo,mean)
plot(c(1:12),Close_mon,type='b',xlab="Mois de l'année",ylab="valeur de cloture",main="Les valeur
s de cloture en fonction des mois")
```




```
boxplot(data$Close ~ mo, col = "lightblue", pch = 20, cex = 0.5,xlab="Mois de l'année",ylab="Val
eur de cloture",main="Les boîtes à moustaches en fonction du mois")
```

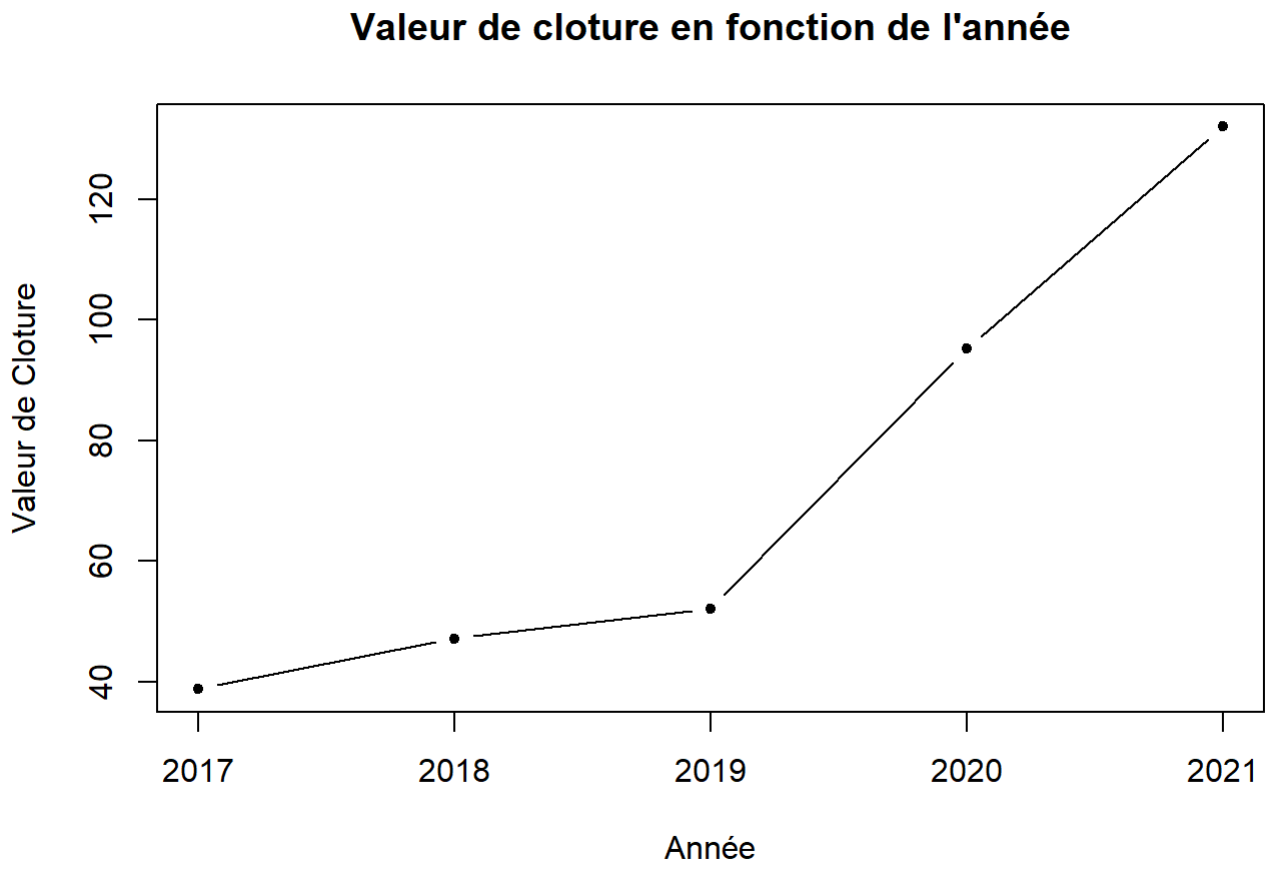


Les valeurs de cours de cloture mensuelles prennent une forme intéressante. On observe un maximum au mois du janvier et fevrier puis une grande diminution vers le mois de mars. ceci est intéressant vu que la diminution est assez grande pour spéculer sur la saisonalité.

Analyse selon les années

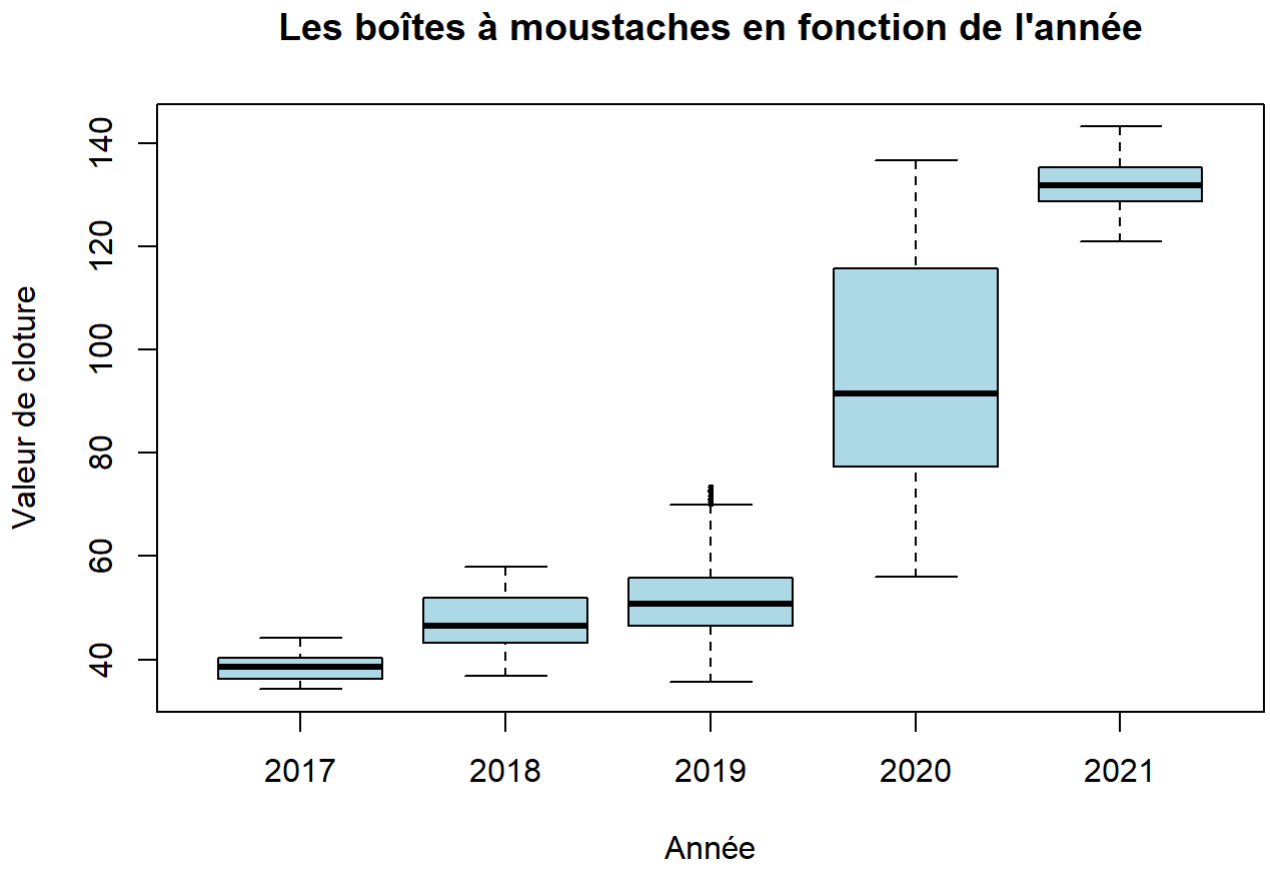
Contrairement aux analyses précedentes, l'analyse selon les années dans notre cas nous permet d'avoir une idée sur la tendance parce que le jeu de données ne couvre que 4 années.

```
Year <- as.factor(format(data$Date, "%Y"))
YearlyClose <- tapply(data$Close, Year, mean)
plot(c(2017:2021),YearlyClose, type = "b", pch = 20,xlab="Année",ylab="Valeur de Cloture",main=
"Valeur de cloture en fonction de l'année")
```



Cette allure montre une croissance accélérée au cours des années où l'acceleration a lieu après 2019

```
boxplot(data$Close ~ Year, col = "lightblue", pch = 20, cex = 0.5,xlab="Année",ylab="Valeur de c
loture",main="Les boîtes à moustaches en fonction de l'année")
```

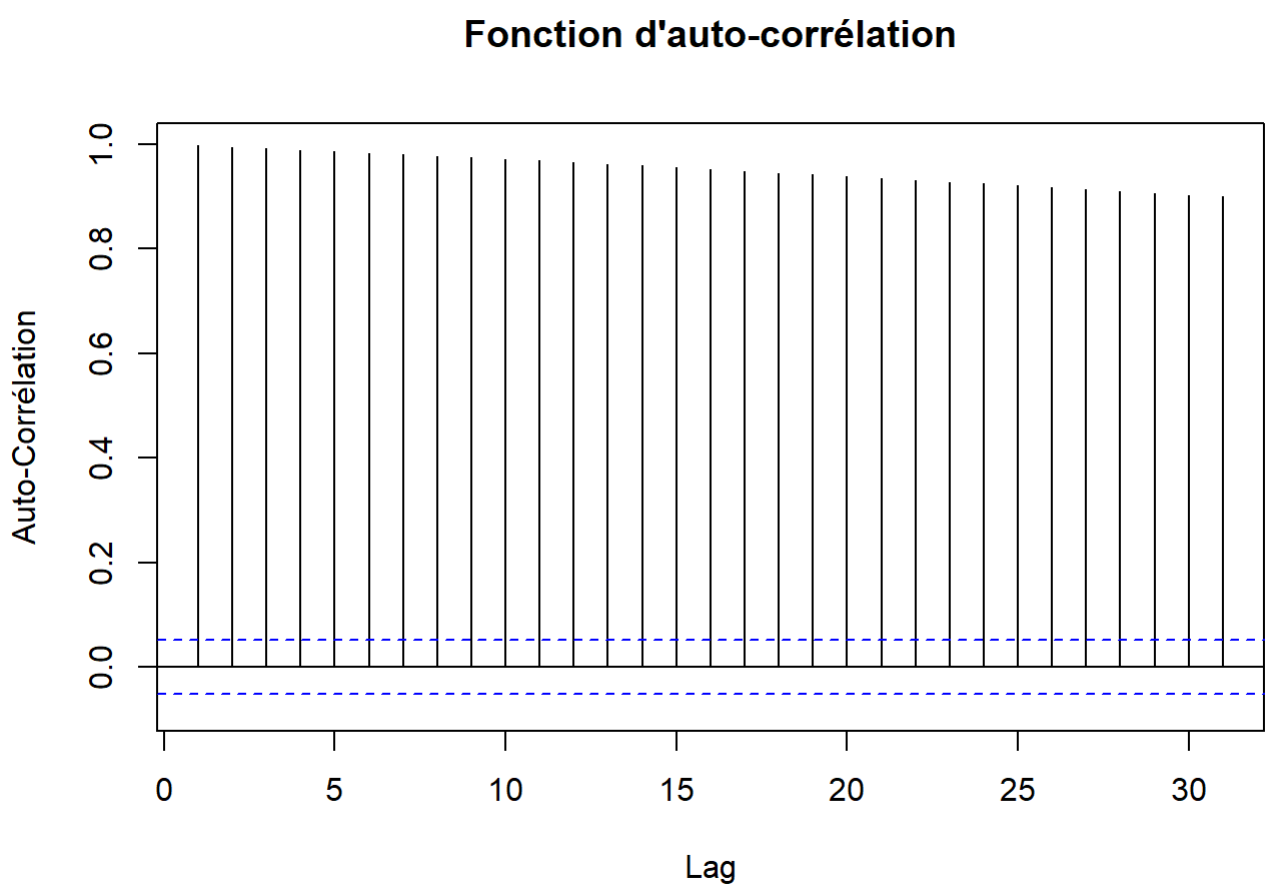


les boites à moustaches des années 2017,2018,2019 et 2021 ont approximativement la même allure où les valeurs sont faiblement dispéesées tandis que le boxplot de l'année 2020 est plus large donc on peut supposer d'après la figure précedente que cette année est une année de transition où la valeur du cours a augmenté drastiquement.

l'estimation de la fonction d'auto-corrélation

On estime, à l'aide de la fonction acf la fonction d'auto-corrélation de notre jeu de données:

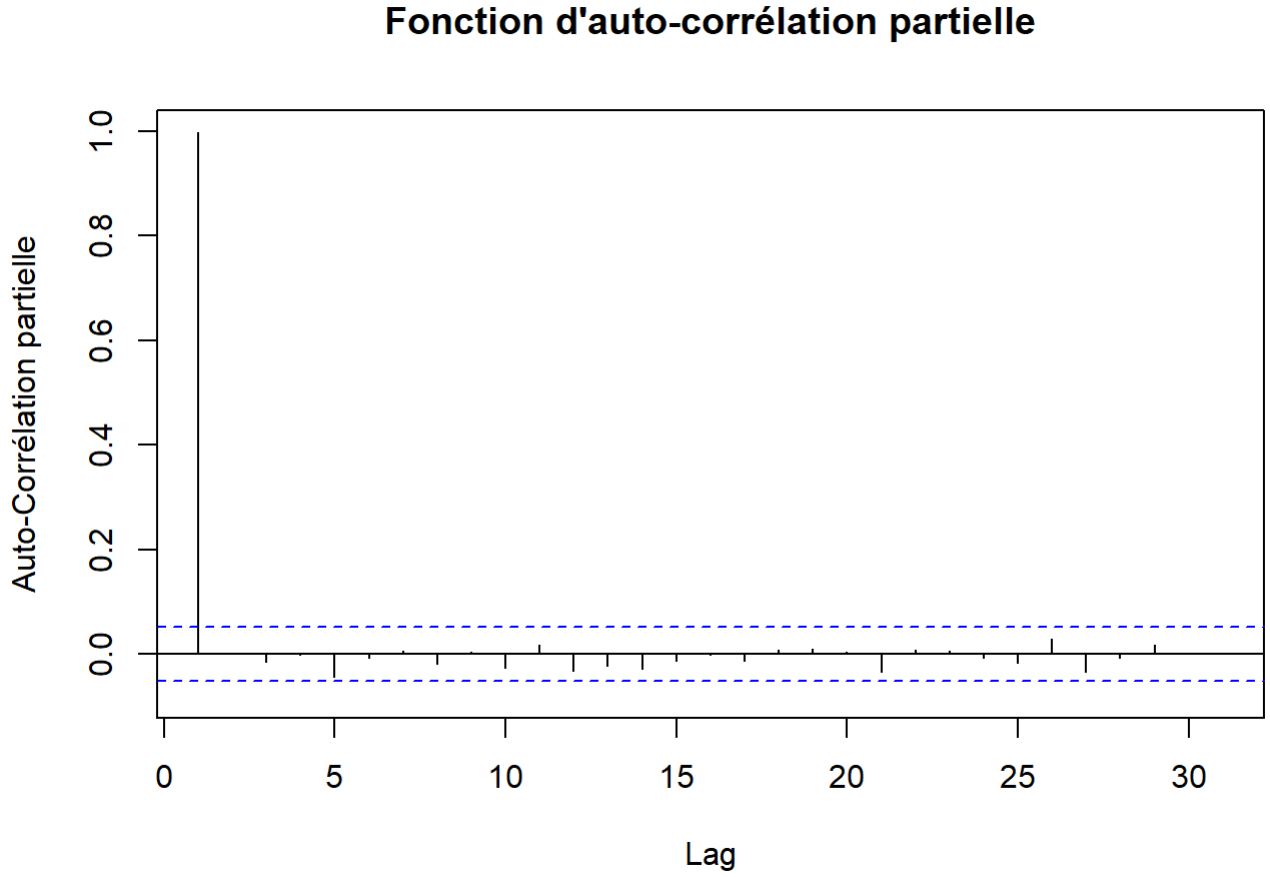
```
Acf(d.xts,main="Fonction d'auto-corrélation",xlab="Lag",ylab="Auto-Corrélation")
```



On observe une forte corrélation entre les instants proches qui est raisonnable vu la nature de la dépendence inhérente des valeurs de clotûres.

Pour éliminer l'effet de cette dépendence, on peut appliquer la fonction Pacf qui calcul l'auto-corrélation entre deux instants en éliminant les dépendances intermédiaires.

```
Pacf(d.xts,main="Fonction d'auto-corrélation partielle",xlab="Lag",ylab="Auto-Corrélation partie
lle")
```



On observe que la fonction Pacf élimine la dépendence des valeurs intérmediaire pour avoir une distribution d'auto-corrélations qui ressemble à celle d'un bruit.

Modélisation des données

Dans cette partie on modélise la série chronologique pour obtenir un modèle prédictif qu'on testera après sur un échantillon test.

Modèle additif

En observant la série chronologique on remarque qu'elle a une tendance à coître. La composante de faible fréquence est donc facile à extraire. On modélise le signal sous la forme additive $Y_t = T_t + S_t + \epsilon_t$ où T_t: la tendance de la série. S_t: un signal résiduel. ϵ_t: un bruit. En première approximation, on supposera que le signal S_t modélise une saisonnalité dans la série qu'on essayera de la modéliser. En effet, dés la première vue, On remarque que la série présente certains pics distants d'une durée d'un an à peu prés avec des fluctuations mensuelles entre les pics.

Approximation de la tendance

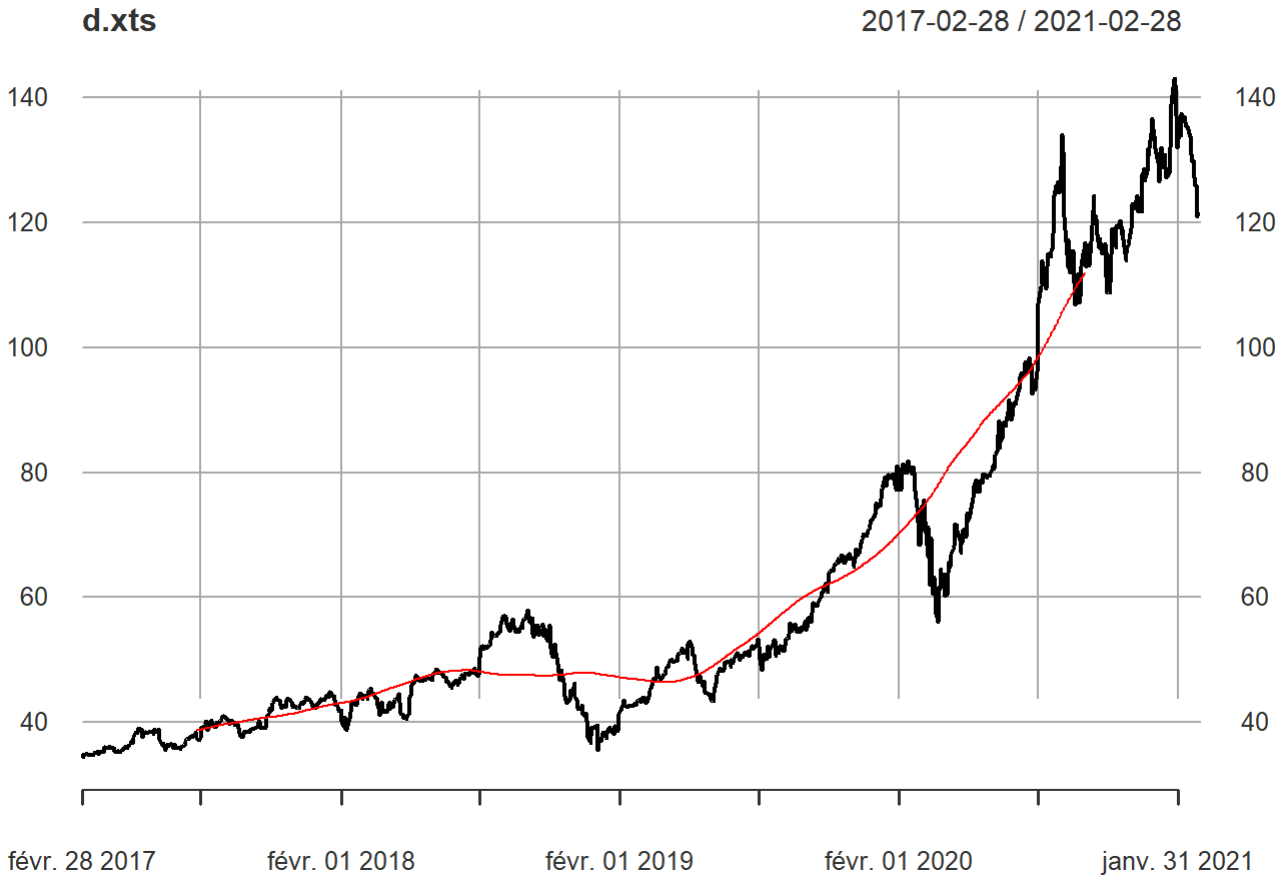
Moyenne Mobile

On utilise ici la méthode de la moyenne mobile avec une fenêtre centrée de 300.

```
MA<-filter(d.xts, filter=array(1/300,dim=300), method = c("convolution"),
           sides = 2, circular = FALSE)

MA<-xts(MA,order.by=data$Date)
```

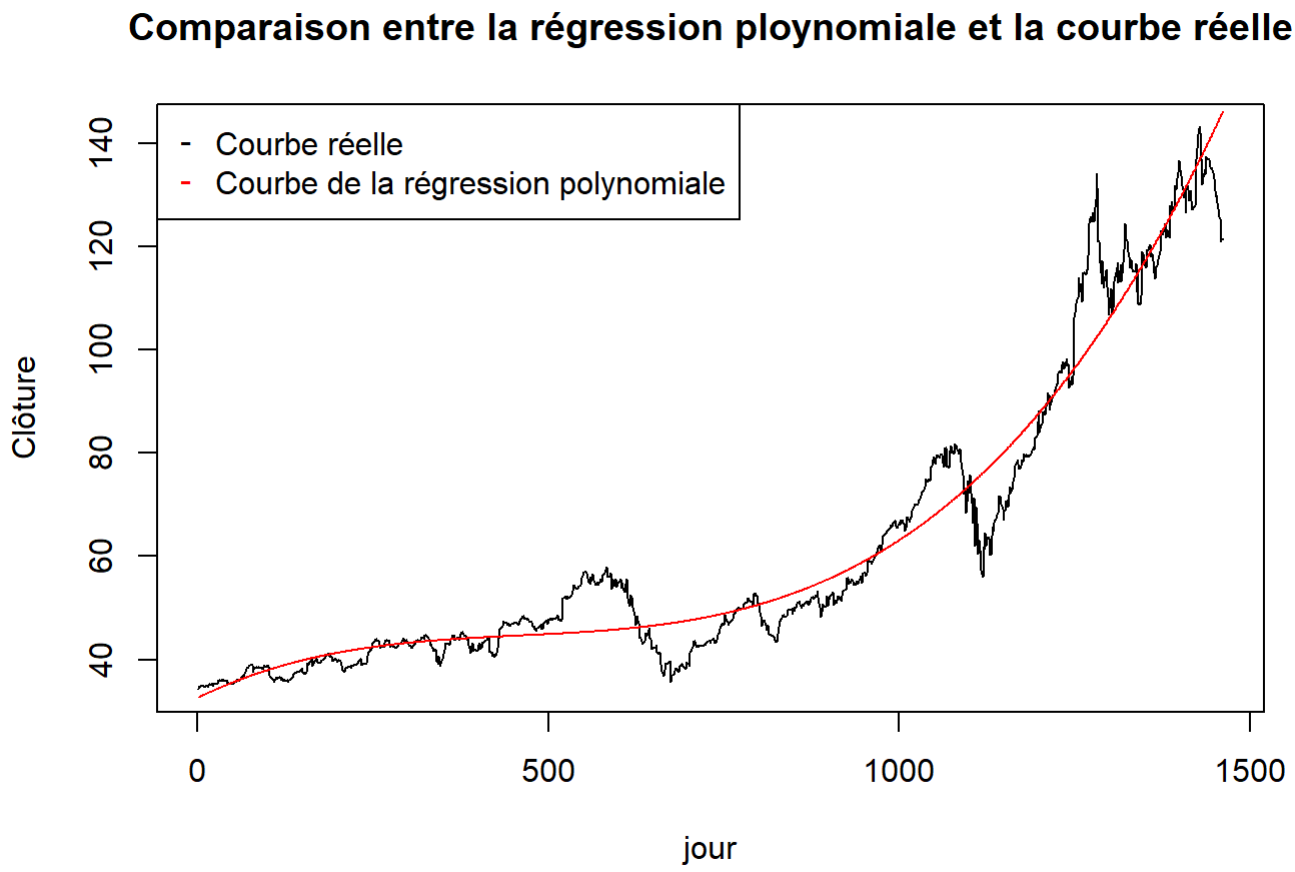

d.xts
2017-02-28 / 2021-02-28
140
120
100
80
60
40
févr. 28 2017
févr. 01 2018
févr. 01 2019
févr. 01 2020
janv. 31 2021


d.xts
2017-02-28 / 2021-02-28
140
120
100
80
60
40
févr. 28 2017
févr. 01 2018
févr. 01 2019
févr. 01 2020
janv. 31 2021

Régression polynomiale

Pour capter les courbures de la tendance on fait une regréssion polynomiale de degré 3 en la variable temporelle.

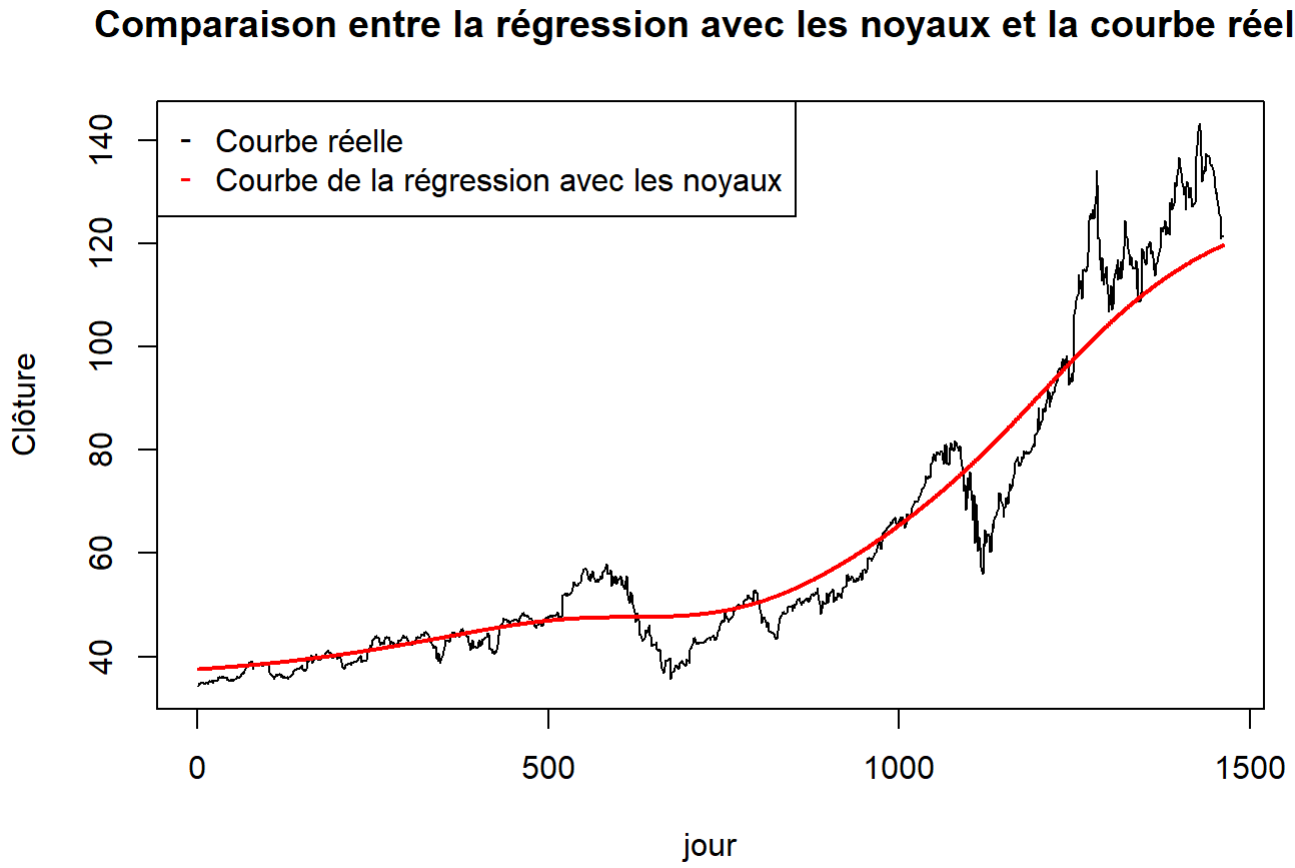
```
time <- c(1:length(data$Date))
reg <- lm(data$Close ~ time + I(time^2)+I(time^3), data=data)
```



Noyau Gaussien

On fait passer ici un noyau gaussien sur la série pour modéliser sa tendance.

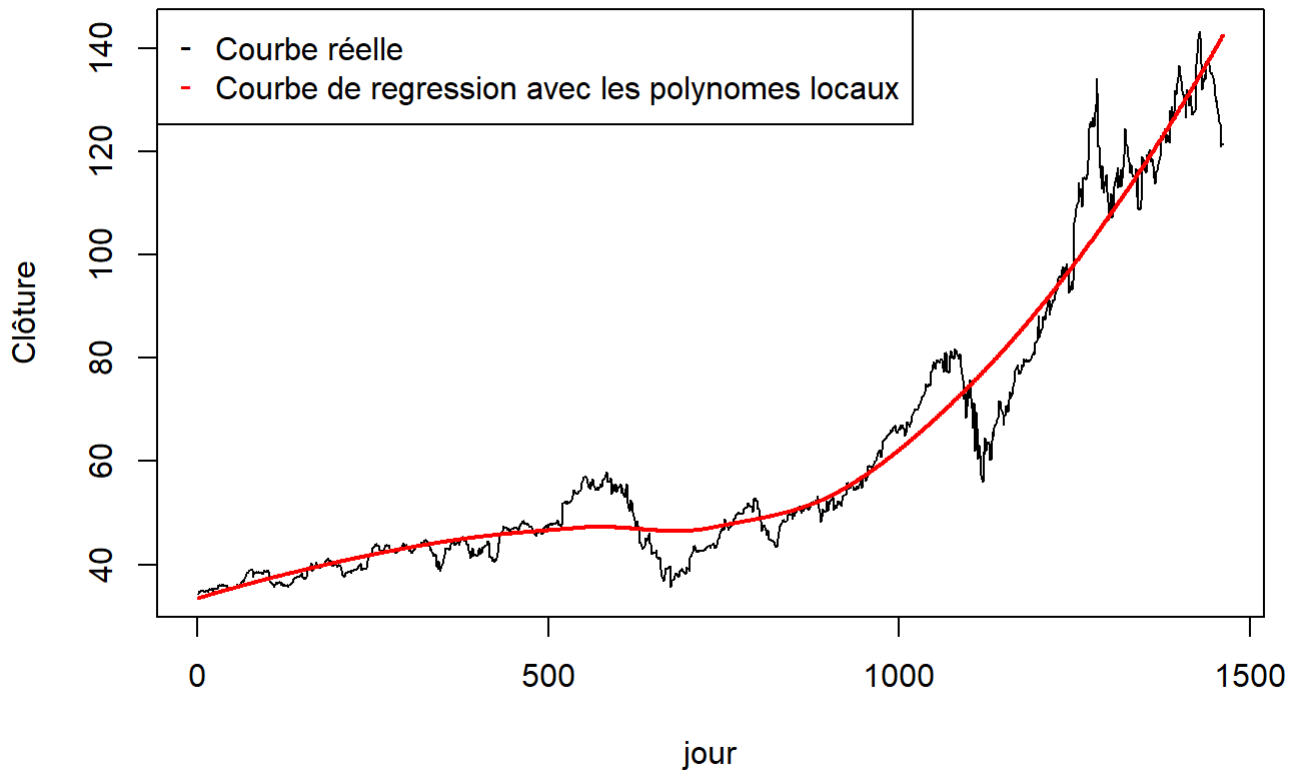
```
noyau <- ksmooth(time, data$Close,kernel = c("normal"),bandwidth =365 )
```



Polynômes locaux

On modélise ici la tendance en utilisant la méthode des polynômes locaux. L'argument degree=2 c'est pour préciser qu'on ajuste un polynôme de degré 1 ou 2 aux points de la courbe. Et l'argument span=0.7 c'est pour considérer un nombre de points suffisament grand pour bien suivre la tendance de la courbe.

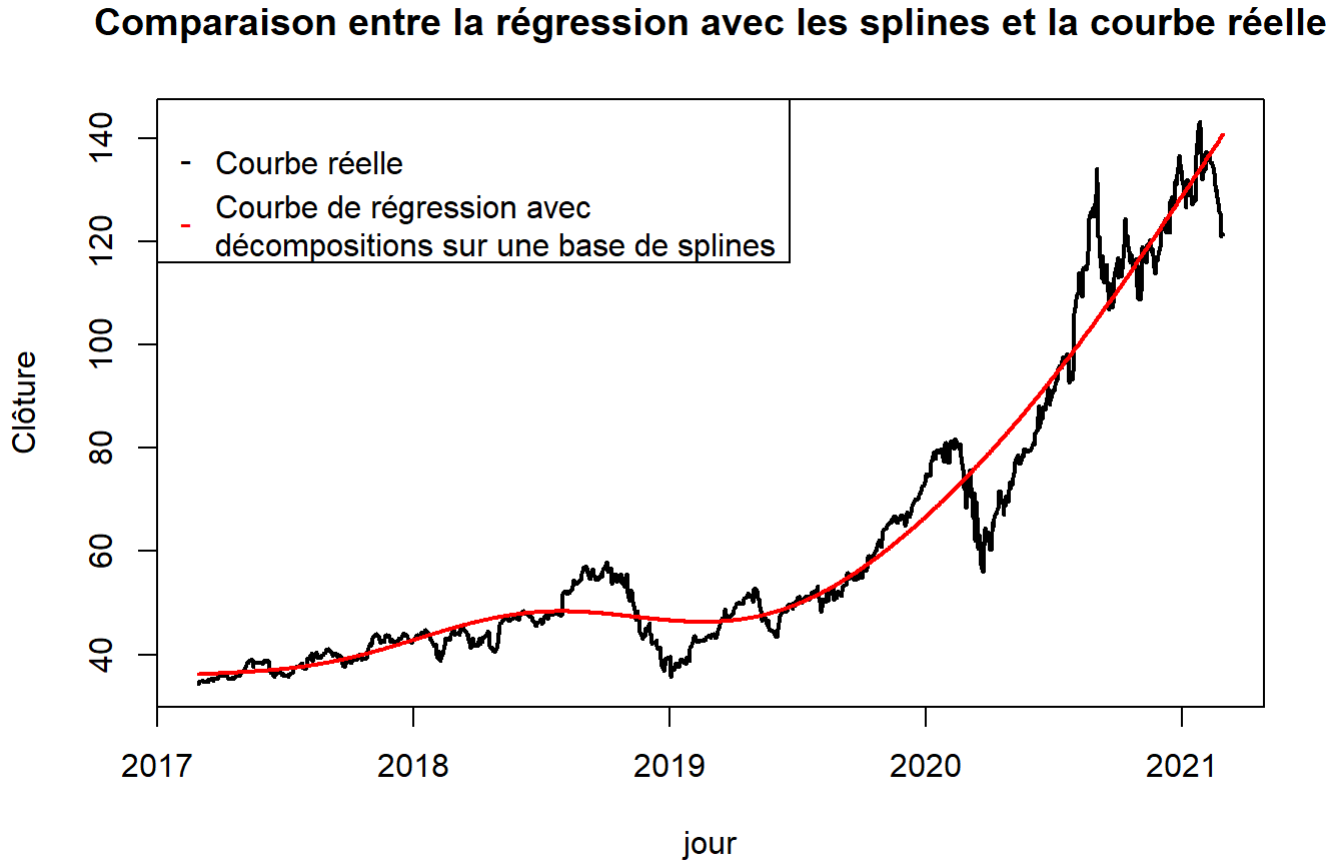
```
lo <- loess(data$Close ~ time, data = data, degree = 2, span = 0.7)
```



Décomposition sur une base de splines

On décompose la série sur une base de splines avec un degré de liberté maximal précisé par k.

```
g <- gam(Close ~ s(time, k = 7), data=data)
```



Signal sans tendance

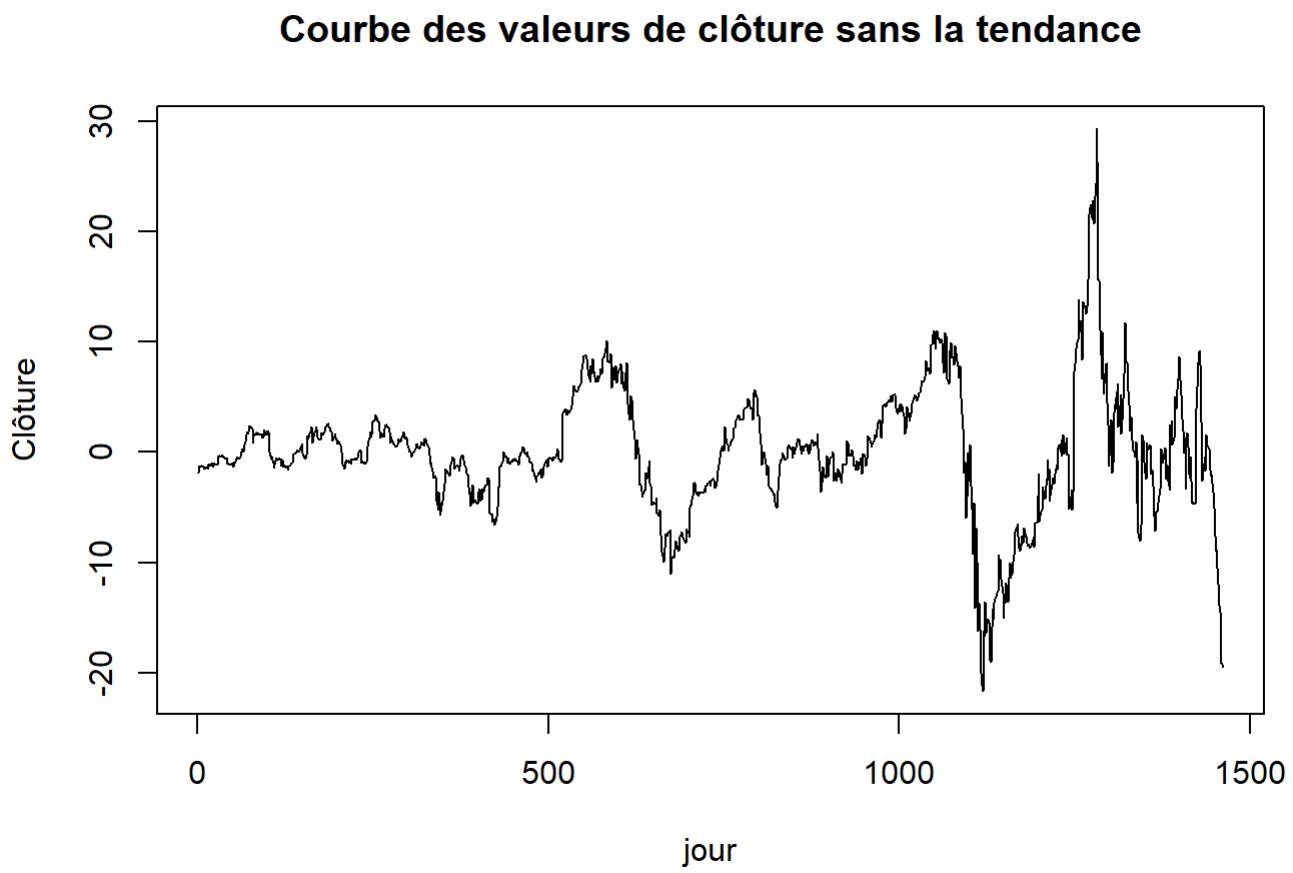
On enlève ici la tendance de notre série et on obsèrve le signal résultant.

```
res=data$Close-g$fitted.values
```

Etude de la saisonnalité

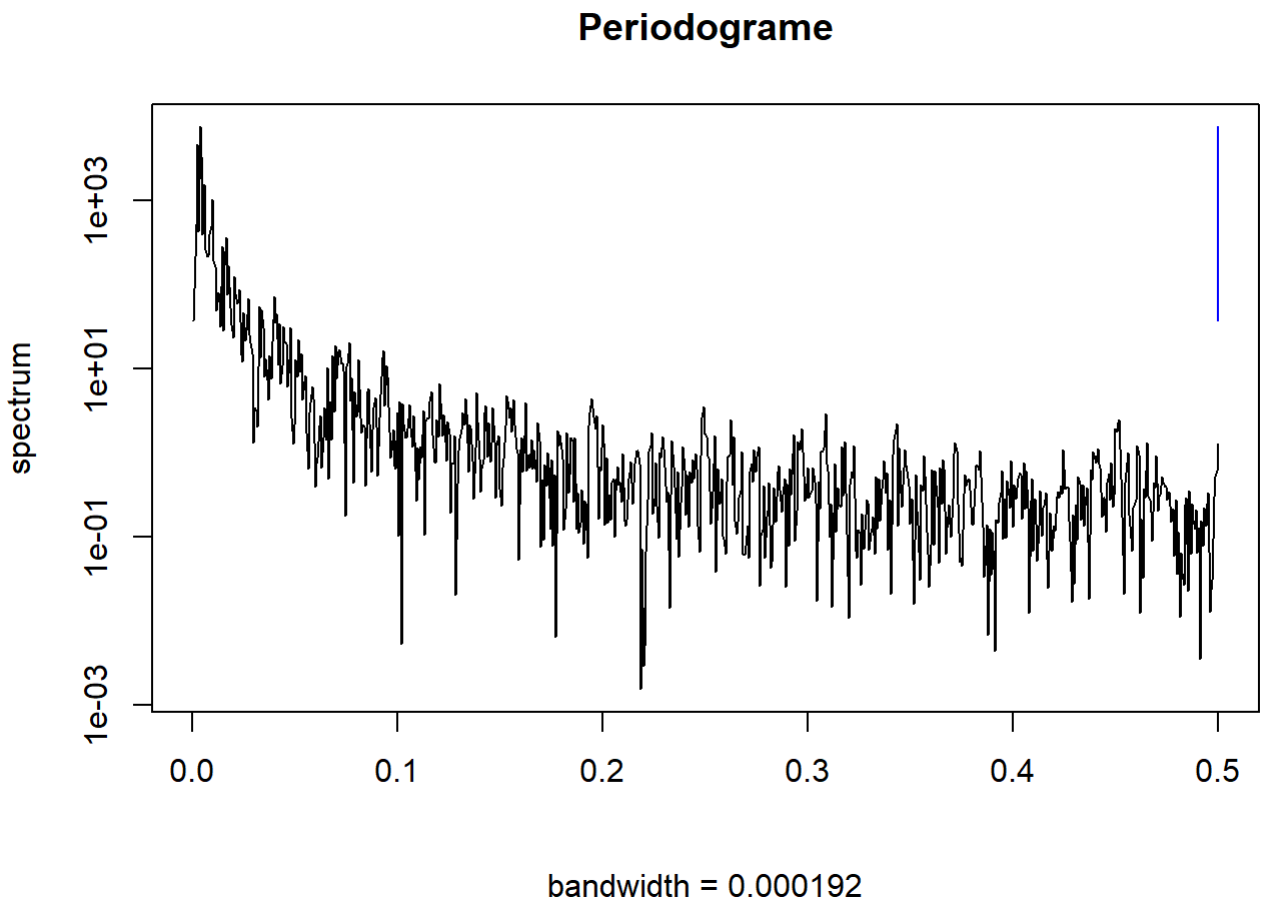
Plot du signal sans tendance

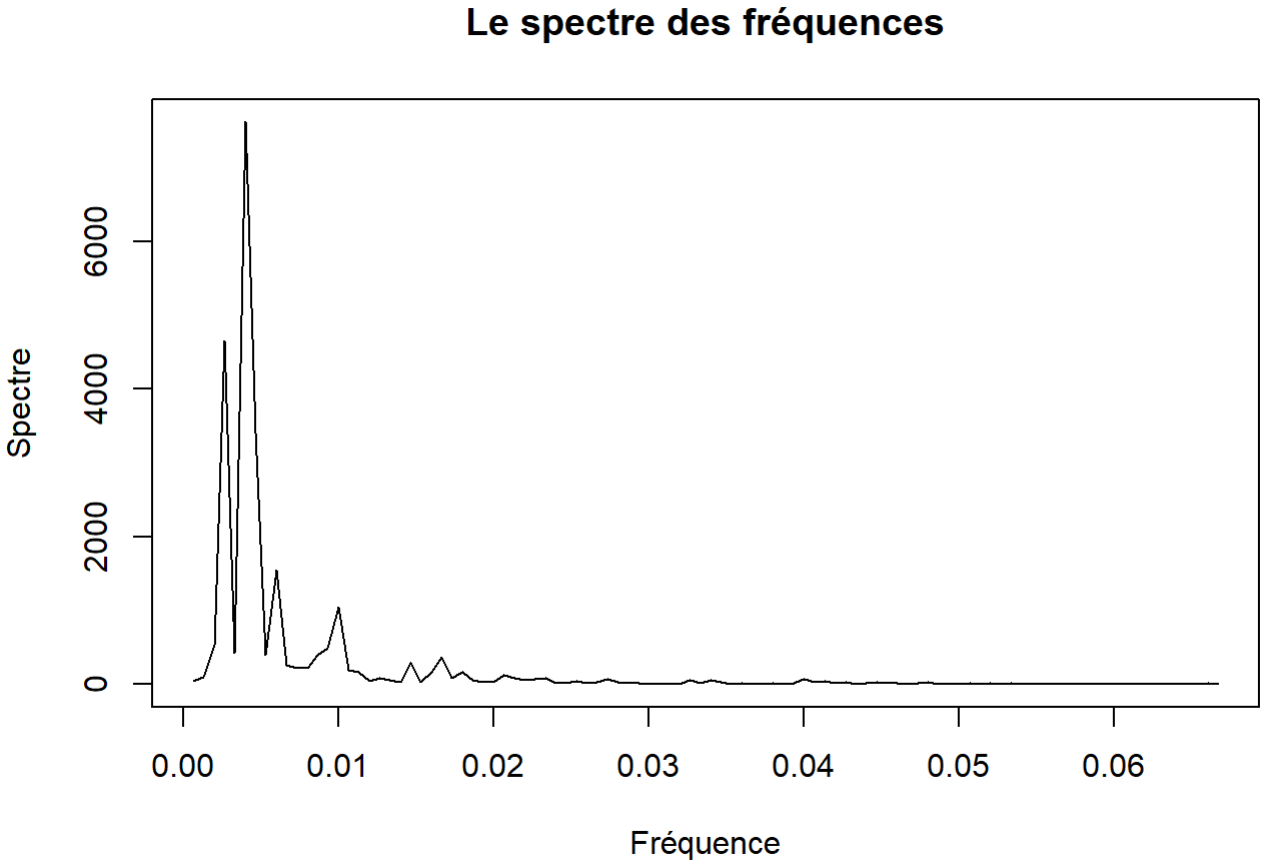
On enlève la tendance de notre signal et on affiche ci-dessous le signal qu'on récupère.



On remarque que le signal obtenu ne présente pas une périodocité remarquable pour modéliser une éventuelle saisonnalité. Pour se convaincre, on peut calculer la densité spectrale du signal résiduel et chercher les fréquences les plus significatives.

Spectre du signal résiduel


Periodograme
spectrum
1e-03
1e-01
1e+01
1e+03
0.0
0.1
0.2
0.3
0.4
0.5
bandwidth = 0.000192


Le spectre des fréquences
Spectre
0
2000
4000
6000
0.00
0.01
0.02
0.03
0.04
0.05
0.06
Fréquence

On remarque que le spectre du signal présente des pics pour des fréquences très faibles dont les amplitudes décroissent rapidement jusqu'à s'annuler. On va prendre les quatres fréquences $f1$, $f2$ et $f3$ correspondant aux pics les plus élevés et on fera une régression sur la base de fourrier constituée par ces harmoniques.

```
a1 = which.max(s$spec)
f1 = s$freq[a1]
a2 = which.max(s$spec[-a1])
f2 = s$freq[a2]
a3 = which.max(s$spec[-c(a1,a2)])
f3 = s$freq[a3]
```

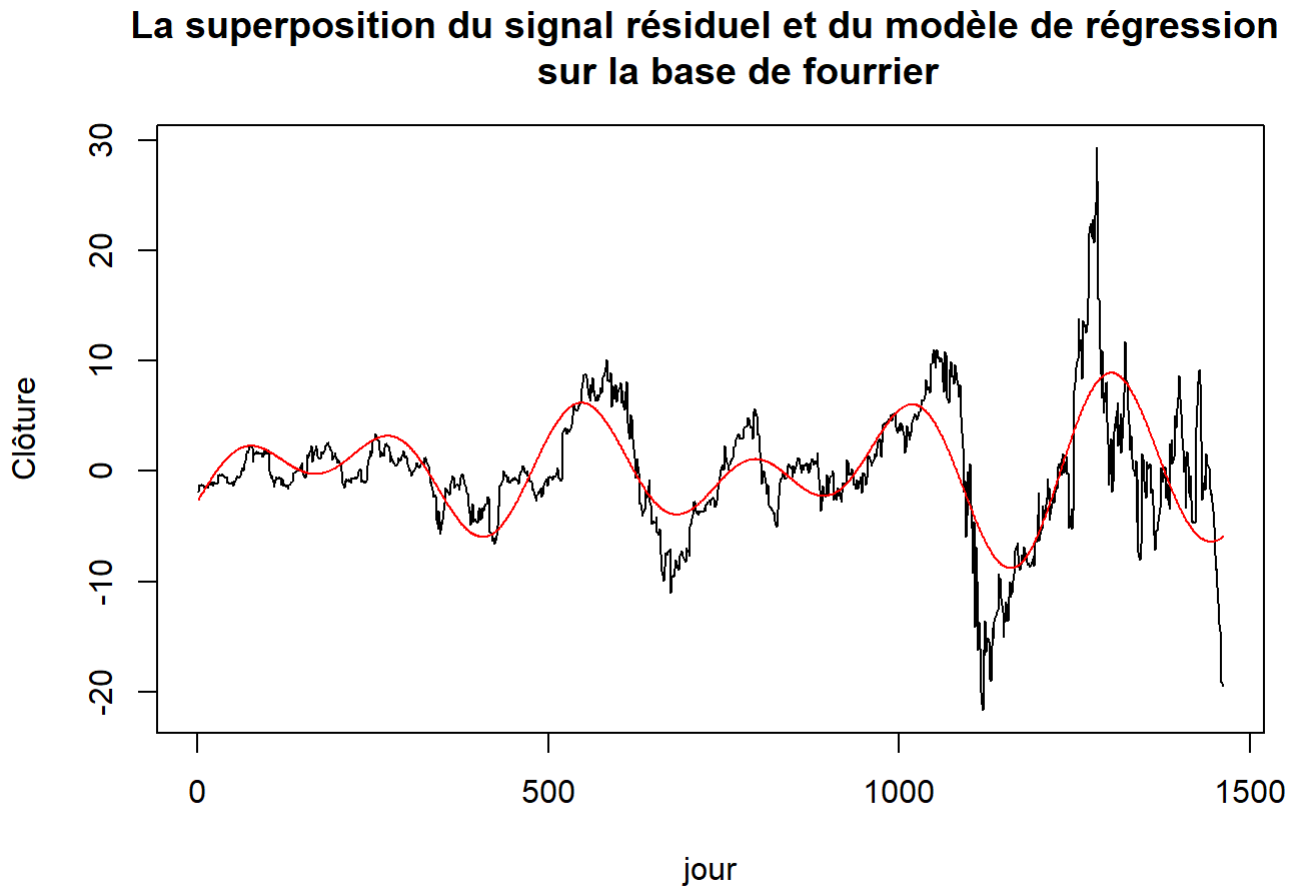
Les trois fréquences correspondantes aux 3 pics les plus élevés sont données ci dessous:

```
## [1] 0.004
```

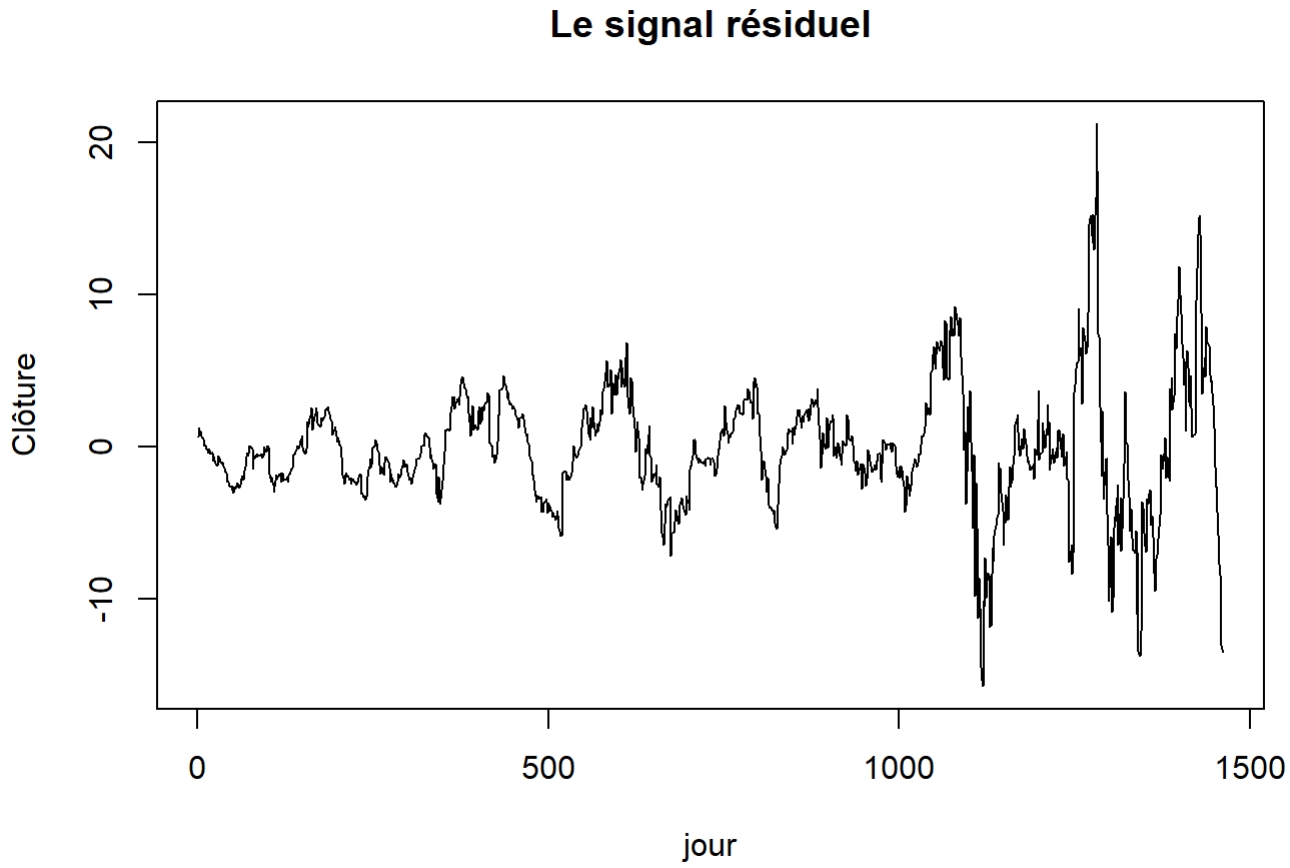
```
## [1] 0.002666667
```

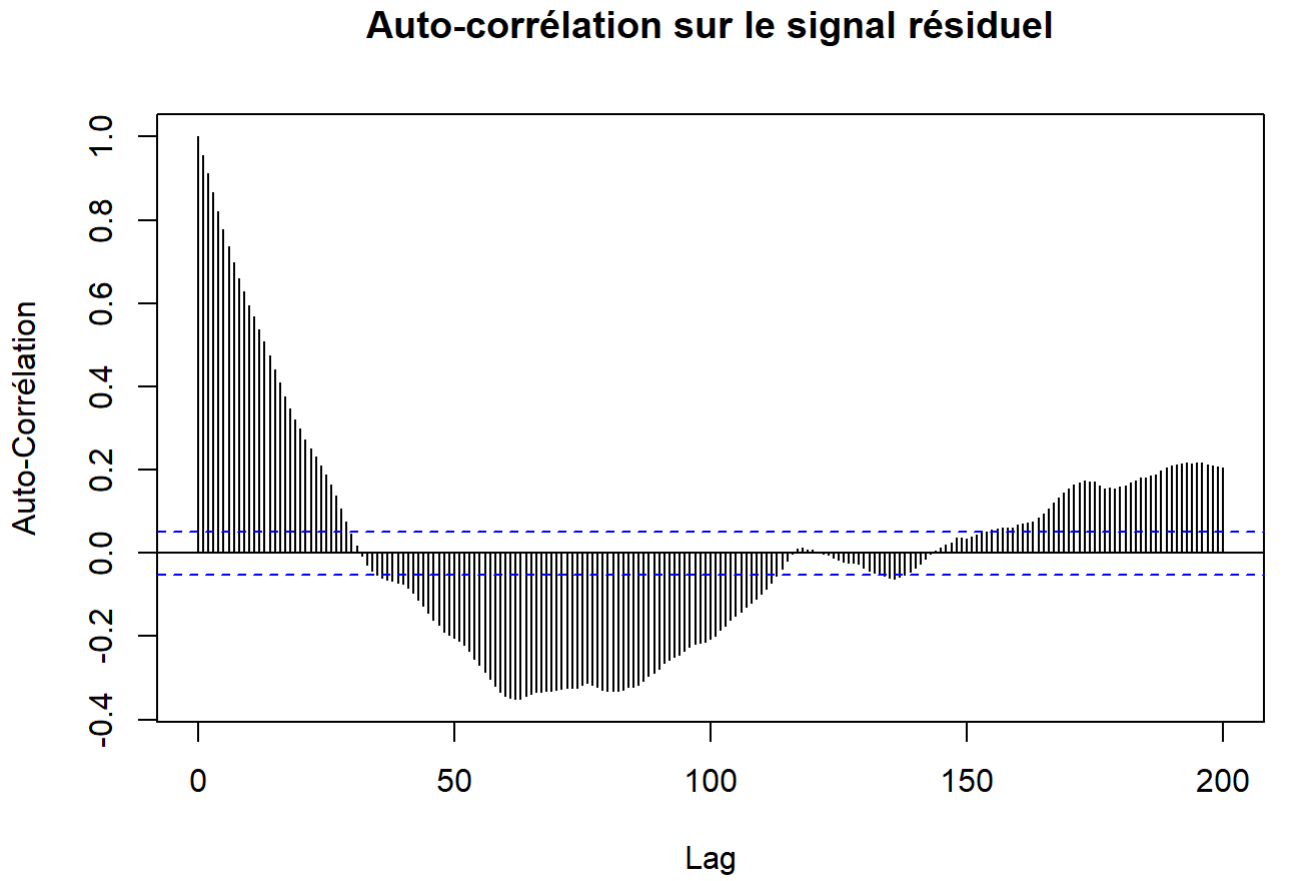
```
## [1] 0.003333333
```

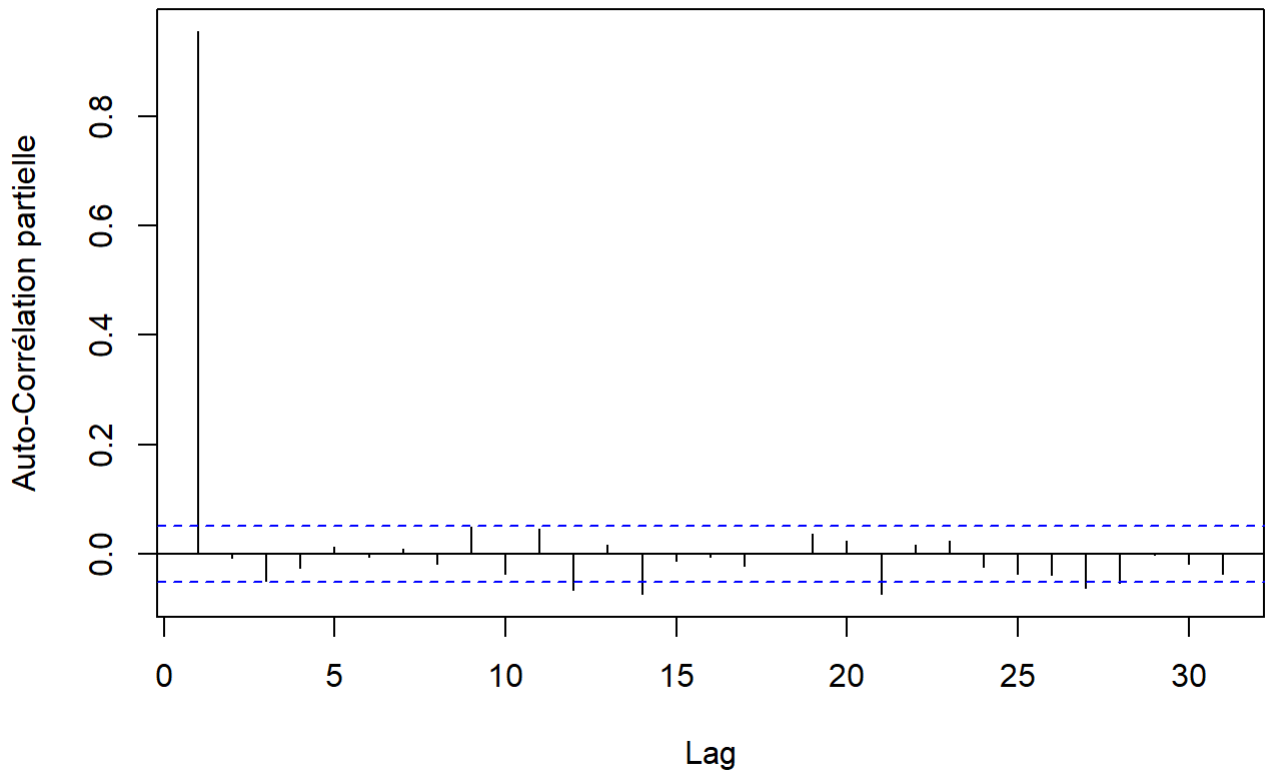
On donne ci-contre la superposition du signal résiduel et du modèle de régression sur la base de fourrier.



On peut remarquer que le modèle de régression arrive à capter en gros les variations du signal. Il est intéressant de voir l'acf et la pacf du signal résiduel après en avoir soustraire le signal du modèle de régression sur la base de fourrier.

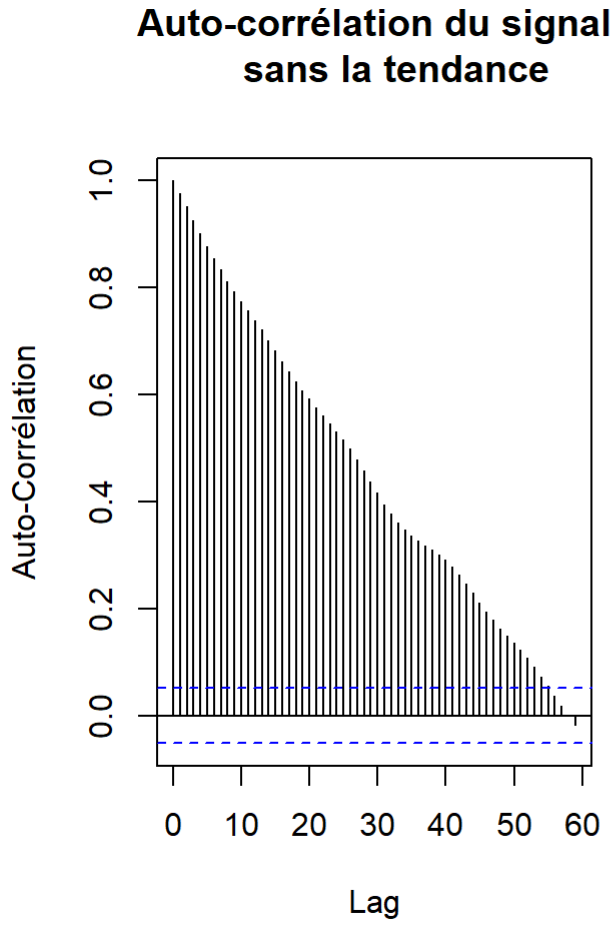

Le signal résiduel
Clôture
20
10
0
-10
0
500
1000
1500
jour

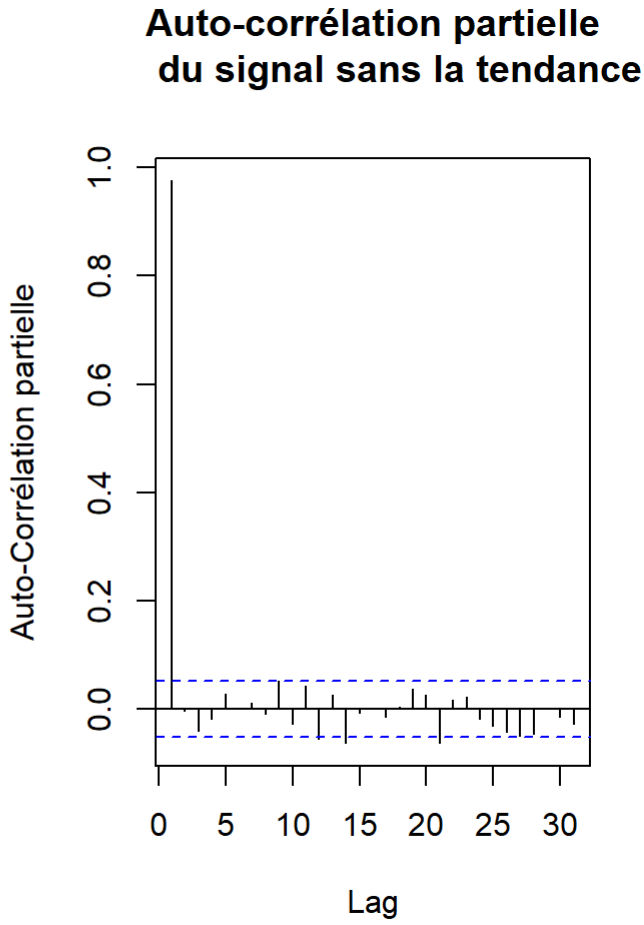

Auto-corrélation sur le signal résiduel
Auto-Corrélation
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
0
50
100
150
200
Lag



En observant l'acf On note la présence de composantes non nulles pour les différentes valeurs du lag. Le signal obtenu à la fin ne correspond pas au modèle additif qu'on a supposé au début. On conclut que la série chronologique ne présente pas une saisonnalité. Cependant, on peut analyser le signal $R_t = Y_t - T_t$ (la série chronologique - sa tendance).

On peut afficher l'acf et la pacf du signal R_t ci-dessous.



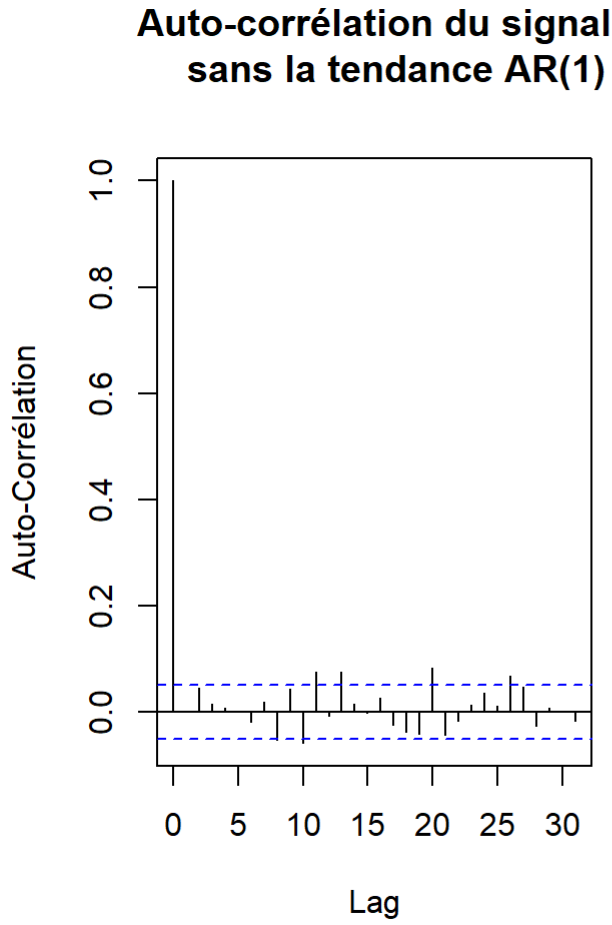


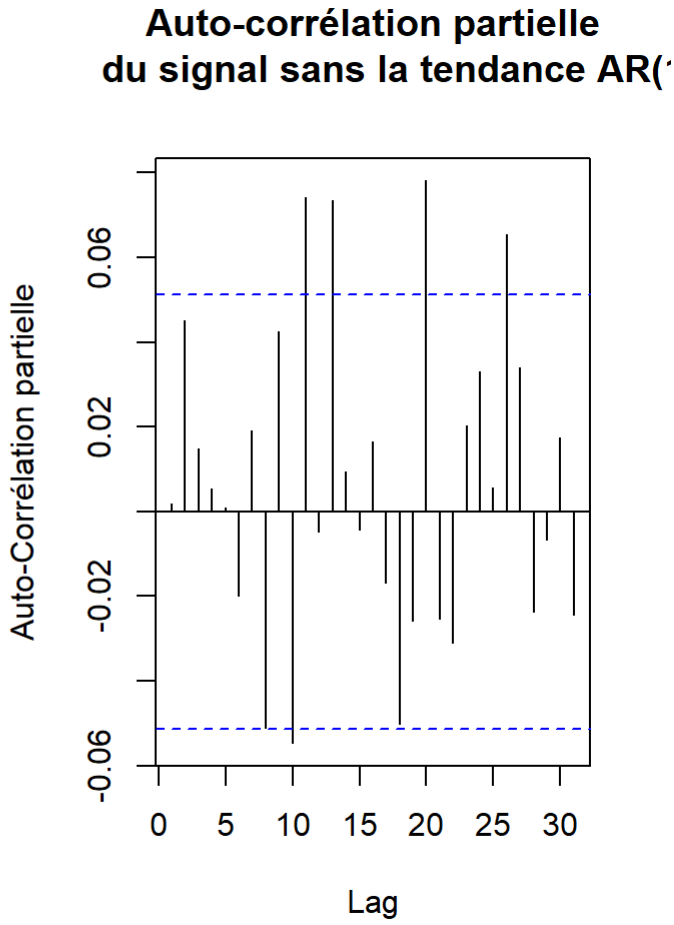
On remarque une décroissance exponentielle dans l'acf du signal R_t et une composante non nulle dans la pacf de lag=1. On peut proposer d'estimer ce signal par un modèle AR d'ordre 1.

Modèle AR de degré 1

```
R.ts = ts(res)
R.model = arima(R.ts, order = c(1,0,0), method = c("ML"),
                SSinit = c("Rossignol2011"),
                optim.method = "BFGS",include.mean = F)
```

On trace l'acf et la pacf du signal R_t auquel on soustrait les valeurs du modèle AR de degré 1.

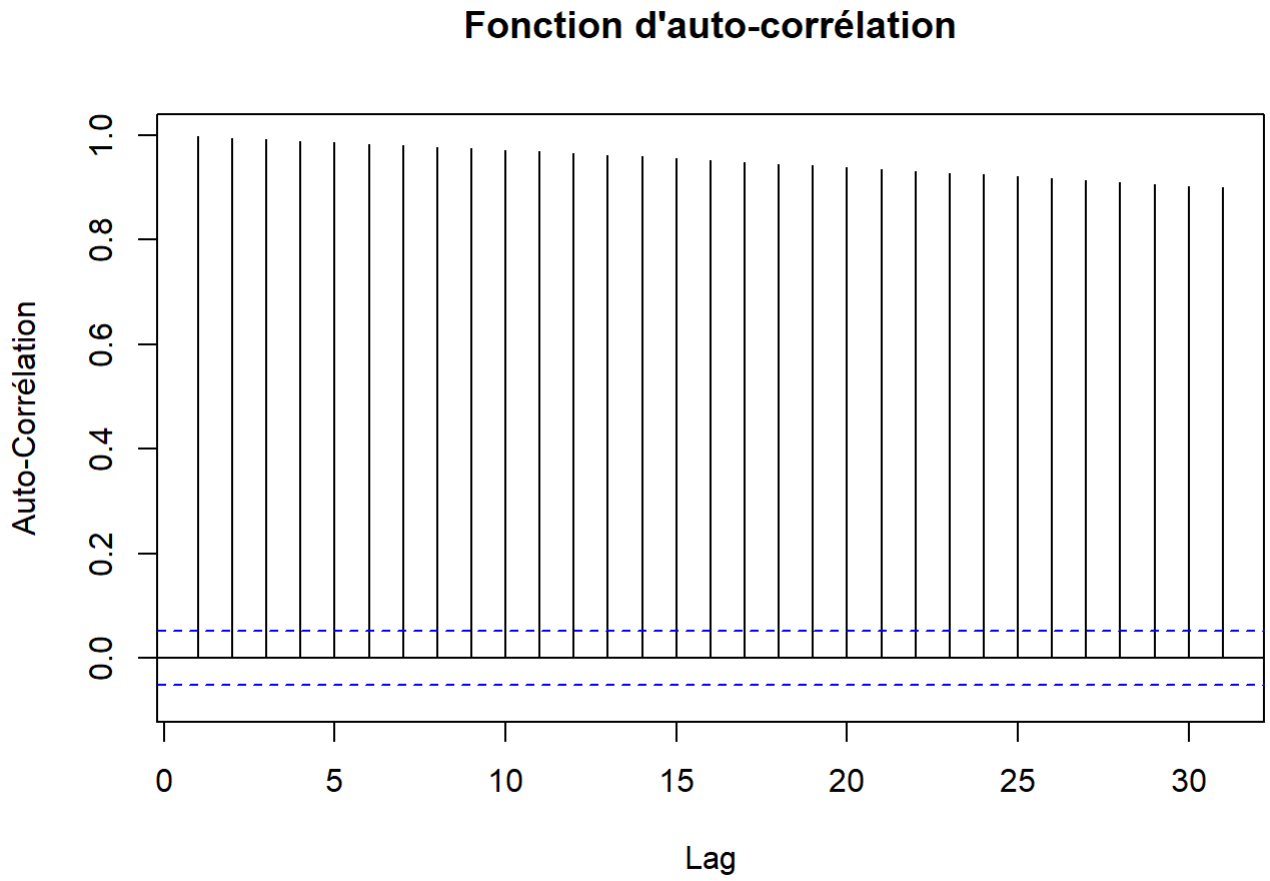




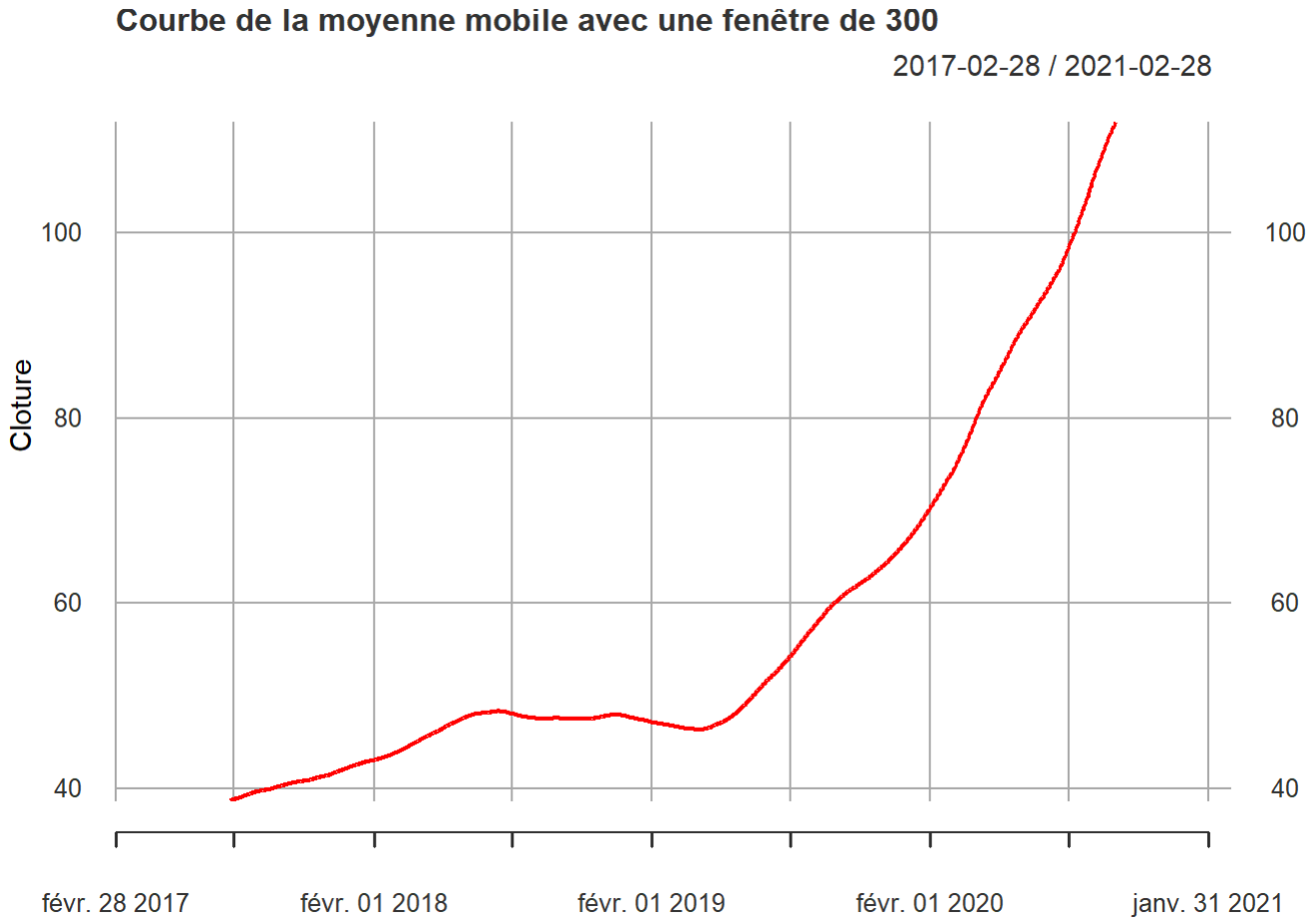
On remarque que le modèle AR de degré 1 est bien adapté à l'approximation du signal R_t. En effet l'acf présente une valeur non nulle au lag=0 et des valeurs presque nulles ailleurs. La pacf confirme aussi ce modèle on remarque aussi des valeurs nulles pour les différents lags en général.

Etude de la stationnarité:

L'allure de la fonction acf nous indique que le processus n'est pas stationnaire vu que les valeurs ne s'annulent pas rapidement en fonction du lag.



Ceci est prévisible vu que l'esperance du processus ne reste pas constante au cours du temps, on peut par exemple effectuer une moyenne mobile pour voir son évolution :



Prévision

Afin de déterminer le modéle de le plus appropié pour faire des prévisions sur notre jeu de données, on va procéder par deux étapes :

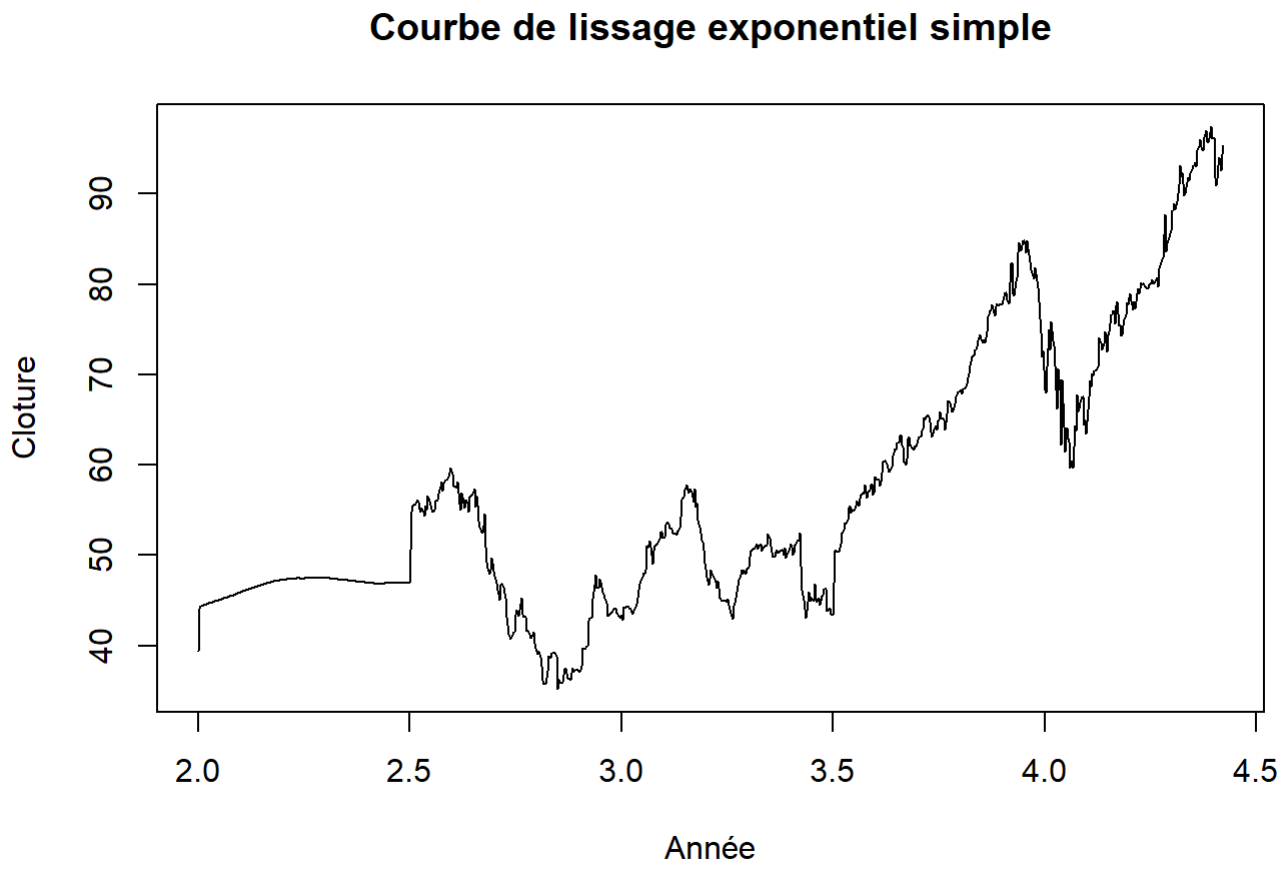
On effectue d'abord une prévision sur des données existantes : On décompose notre base de données en deux : une partie d'apprentissage et une partie test. Les données d'apprentissage sont les valeurs des cours entre 28/02/2017 et 31/07/2020. Puis on extrapole les données du modéle par 212 jours pour atteindre 28/02/2021. On fait ceci pour évaluer l'erreur de prévision de chaque modéle et finalement on extrapole les données sur une autre année (jusqu'à 28/02/2022) en utilisant le meilleur modéle , c'est à dire, celui qui minimise l'erreur de prévision.

Prévision avec le lissage exponentiel

Prévision avec lissage exponnentiel simple

on effectue d'abord un lissage exponentiel simple sur les données entre 28/02/2017 et 31/07/2020 avec la méthode HoltWinters en prenant le paramétre beta=FALSE (pour avoir un lissage simple) :

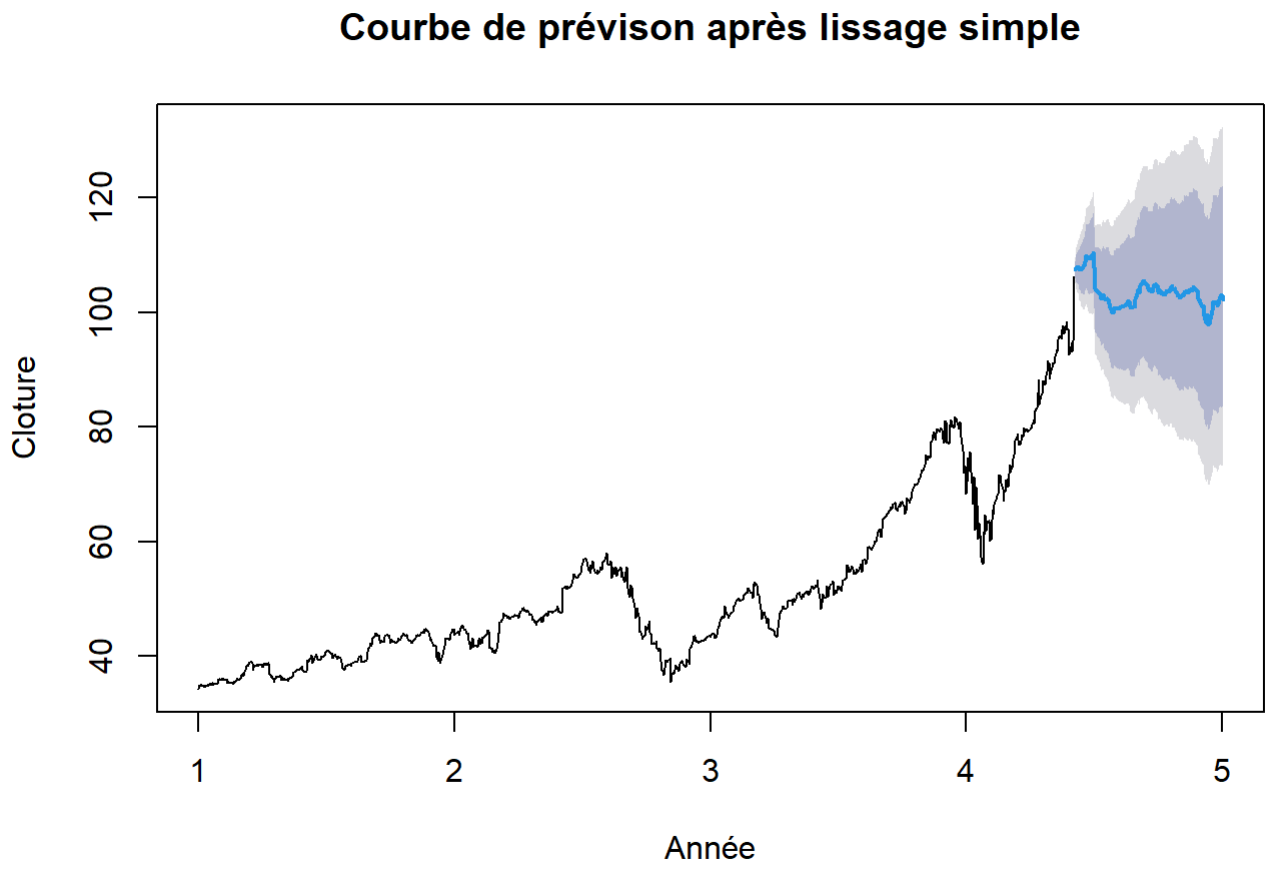
La courbe obtenue avec un lissage simple est la suivante :



On remarque que la courbe après lissage ressemble la courbe sans lissage, d'ailleur, le paramétre α utilisé par holtwinters est très proche de 1 ce qui fait que le lissage prend en consideration la majorité des données :

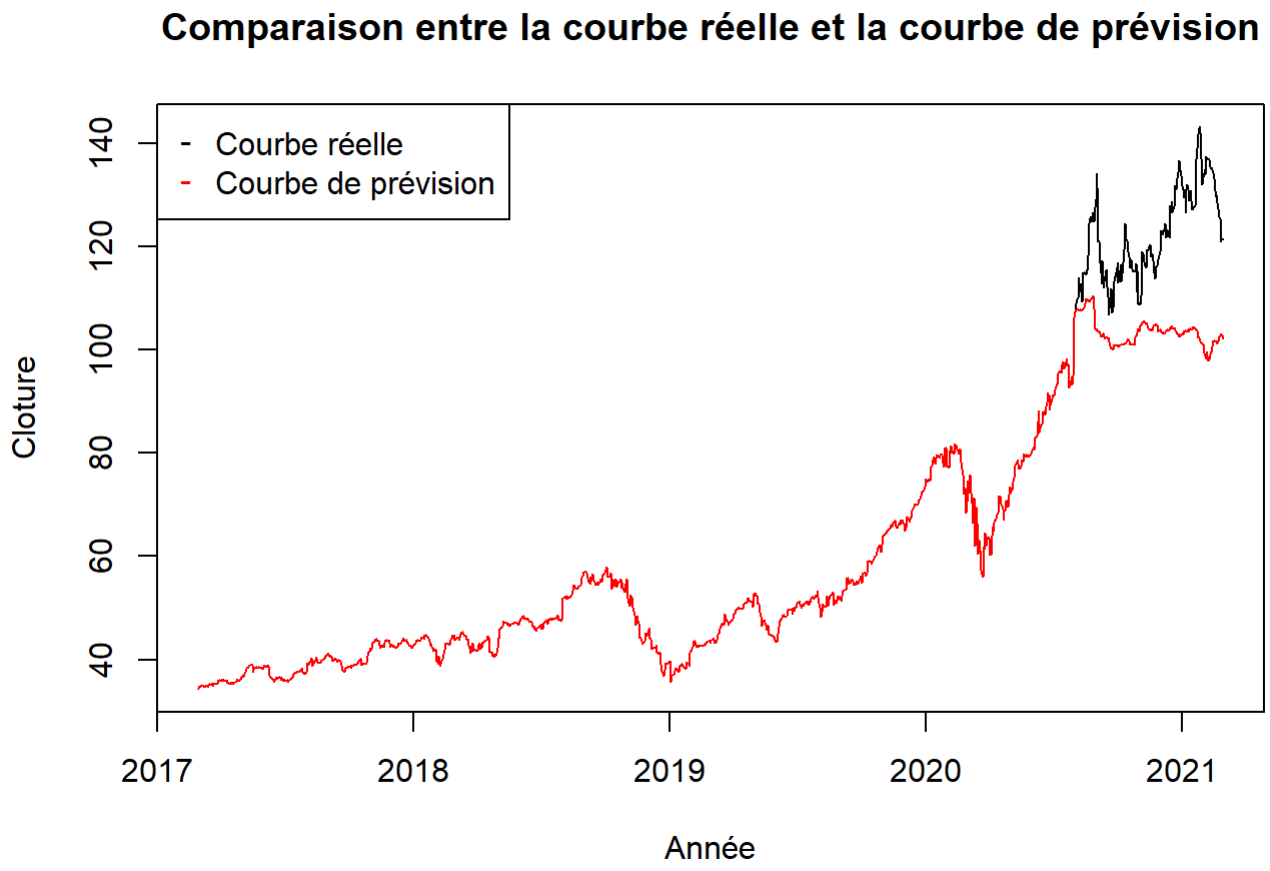
```
##     alpha
## 0.9507794
```

On applique maintenant la fonction forecas.HoltWinters sur 212 jours :



C’est une courbe de prévision avec deux régions de confiance (80% et 95%) où la valeur moyenne des prévision est en trait bleu.

On superpose la courbe réel avec la courbe de prévision et on calcule la norme de l’erreur :



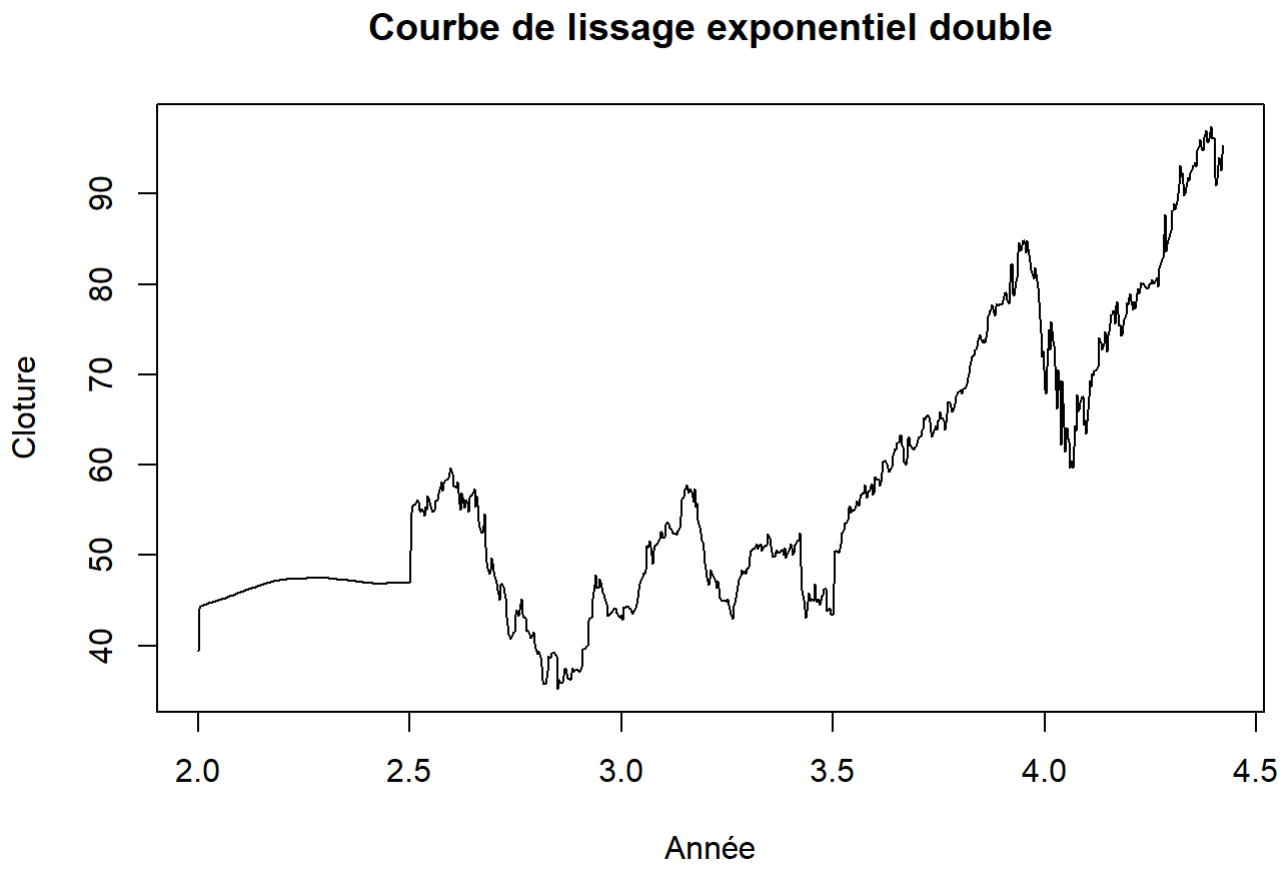
Norme de l'erreur:

```
## [1] 309.1055
```

Prévision avec lissage exponnentiel double

On applique maintenant les mêmes étapes sur le lissage double

La courbe obtenue avec un lissage double est la suivante :

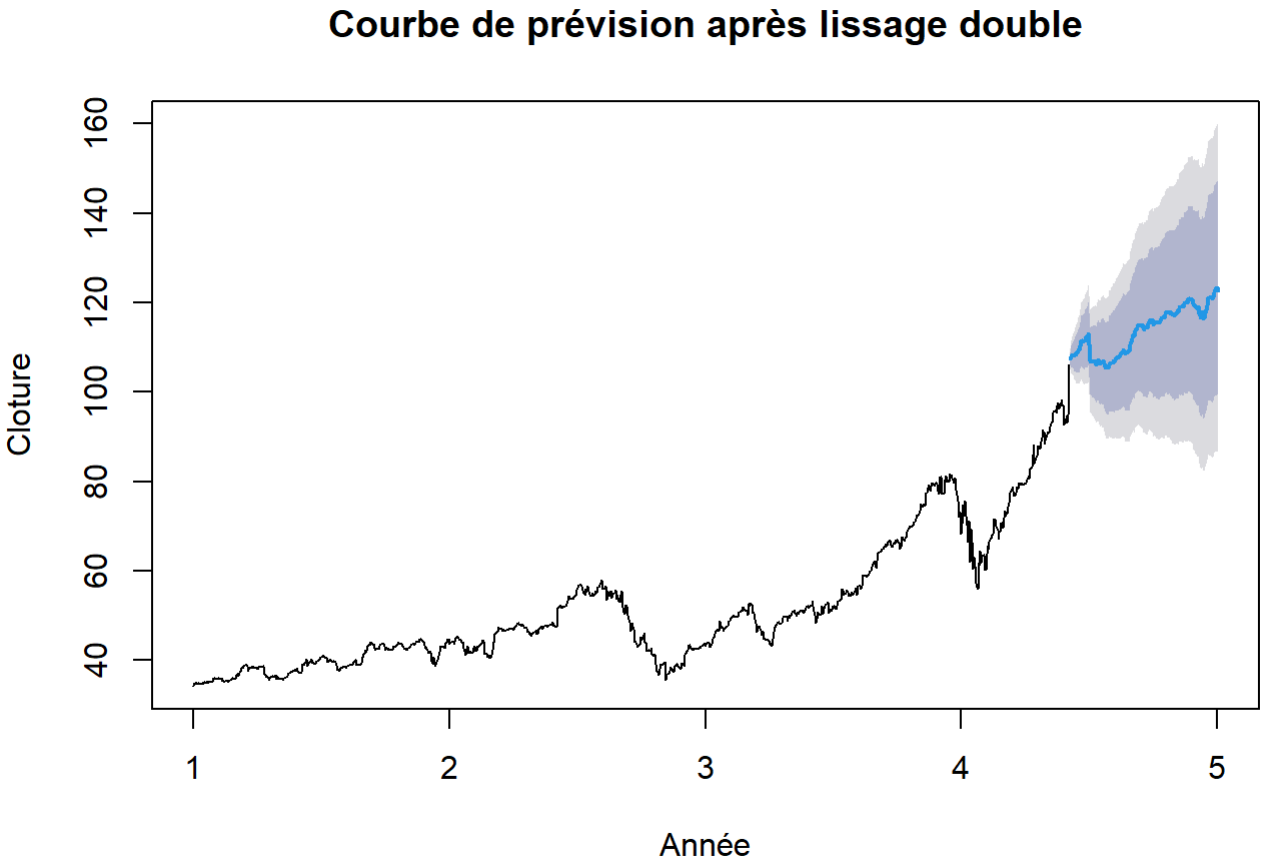


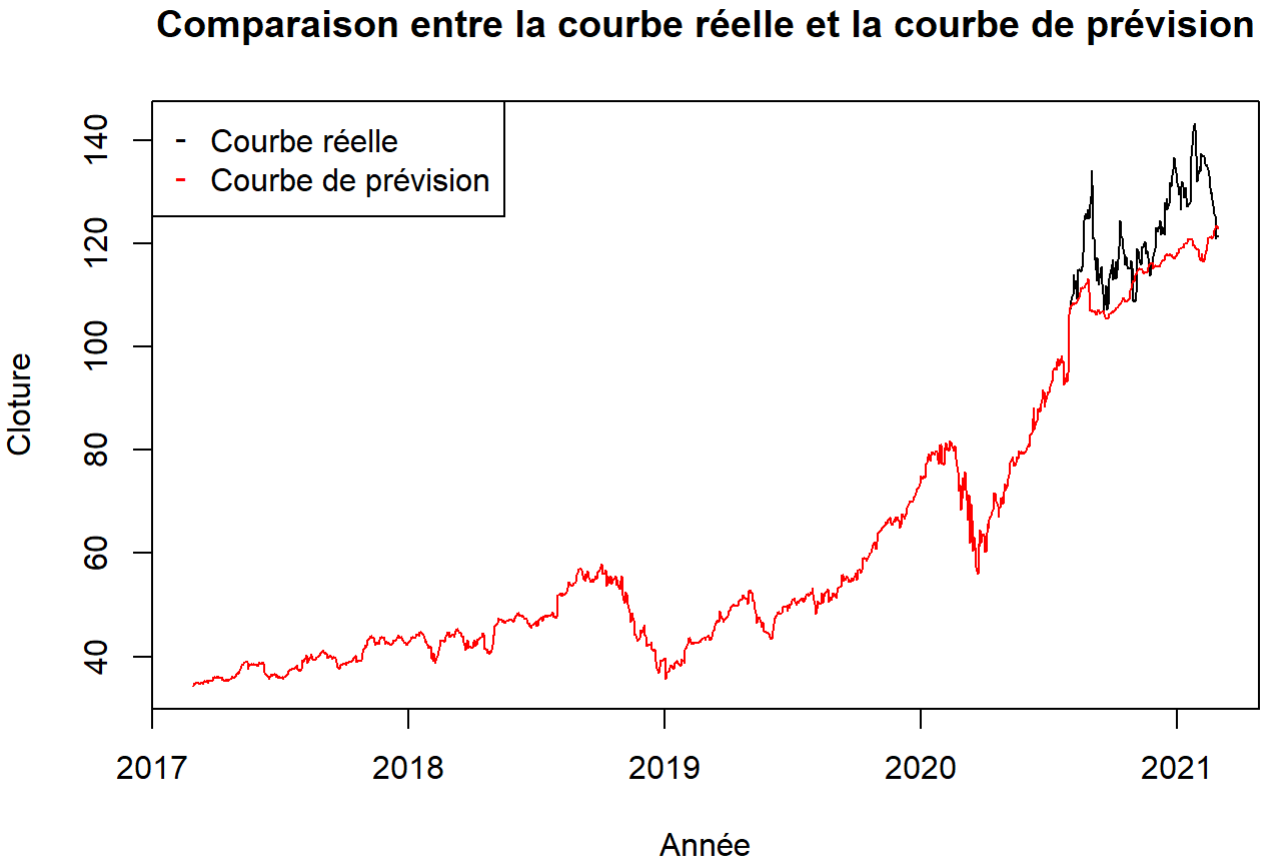
C'est une courbe très proche de celle du lissage simple, et la valeur du paramétre β très proche de 0 le confirme. Les valeurs de α et β pour ce lissage sont les suivantes:

```
##     alpha
## 0.9474153
```

```
##        beta
## 0.002261177
```

On applique maintenant la fonction forecas.HoltWinters sur 212 jours et on superpose la courbe réel avec la courbe de prévision :


Courbe de prévision après lissage double
Cloture
40
60
80
100
120
140
160
1
2
3
4
5
Année


Comparaison entre la courbe réelle et la courbe de prévision
Courbe réelle
Courbe de prévision
Cloture
40
60
80
100
120
140
2017
2018
2019
2020
2021
Année

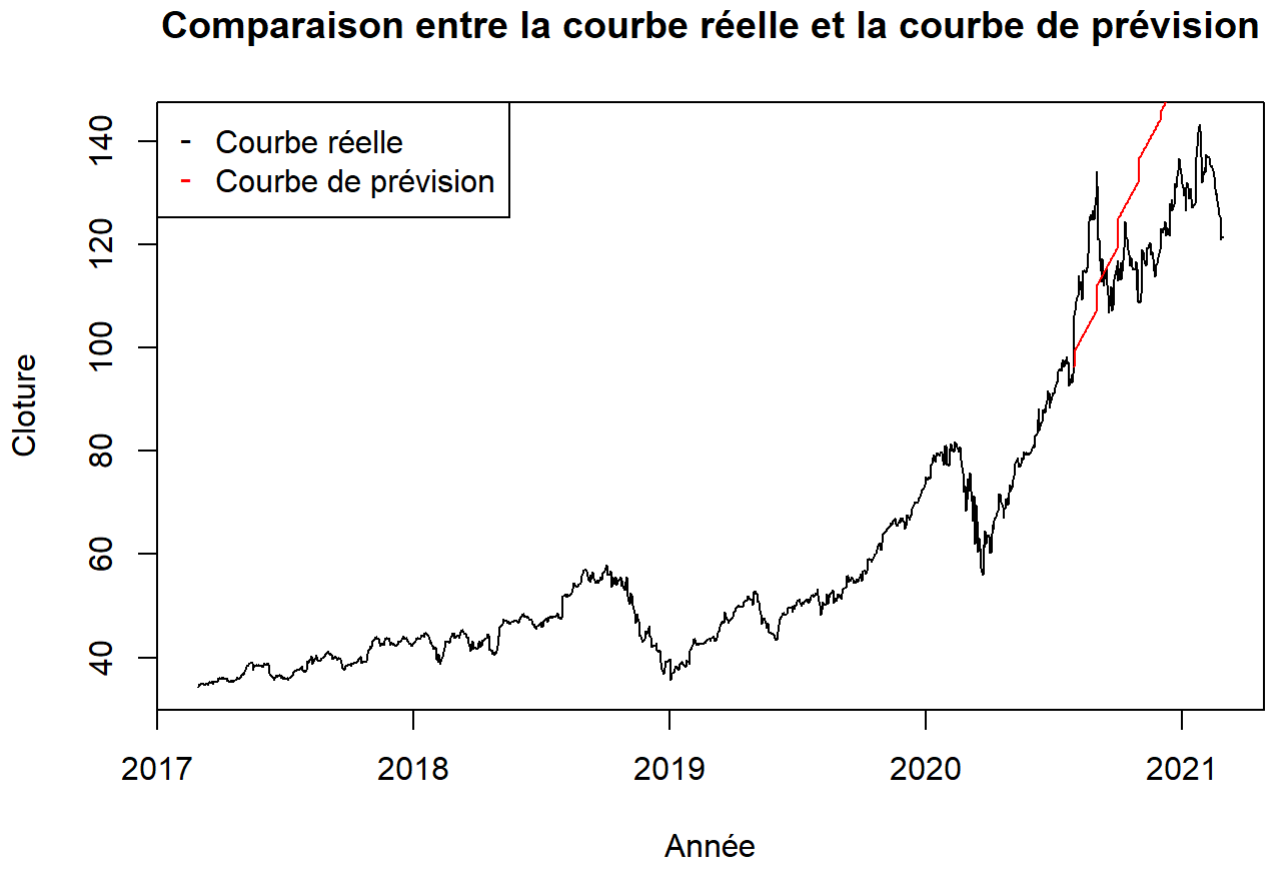
La norme de l'erreur est :

```
## [1] 154.223
```

On observe que la courbe après le lissage double est plus proche que la valeur réelle qui est en concordance avec la norme de l'erreur. Ainsi, le lissage double est préferable ici pour faire des prévisions.

Prévision avec la fonction predict

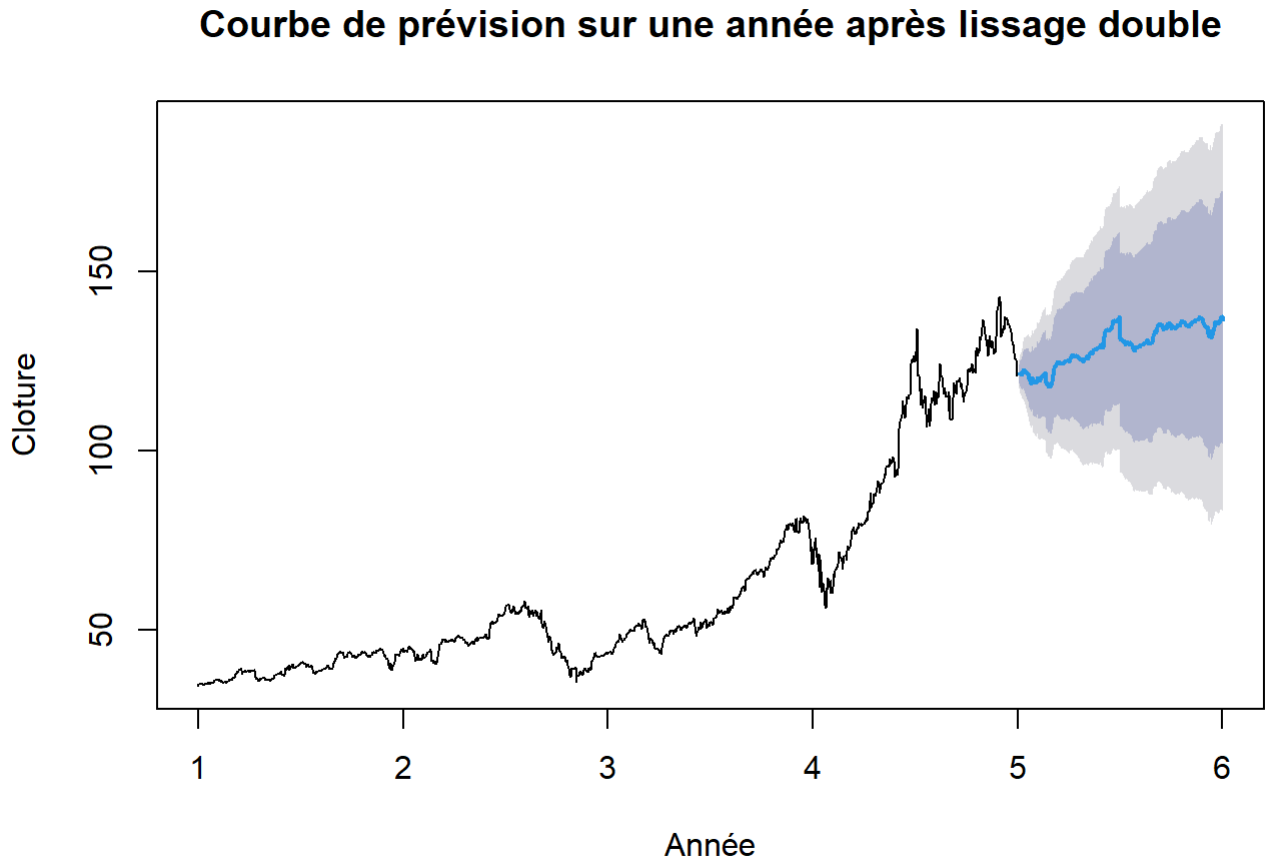
On peut effectuer aussi des prévisions avec la fonction predict ce qui donne l'allure suivante après décomposition sur la base des splines :



Donnant une norme d'erreur égale à

```
## [1] 305.9916
```

De ces trois méthodes, la provision avec le lissage exponentiel double a donné l'erreur la plus faible, on effectue donc une prévision à une année selon cette méthode



La courbe de prévision obtenue est croissante malgrè qu'au moment de l'application de ces provisions la courbe était décroissante. Ceci est dû à la croissance globale de la courbe qui est transmise aux prévisions. ## Conclusion En adaptant un modèle additif pour notre jeu de données, on a pu estimer la tendance générale de la série, sa stationnarité et faire des prévisions sur des valeurs futurs. Cependant, en analysant le périodogramme de la série, on a pas pu détecter une périodicité dans les données ce qui nous amène à dire que la série ne présente probablement pas de saisonnalité. Après l'analyse du signal résiduel R_t, on a opté pour un modèle autorégressif de degré 1 qui donne une modélisation plus fidèle à la série d'origine.

Les résultats de notre analyse peuvent servir pour évaluer la performance d'une entreprise que ce soit d'un point de vu d'investisseurs ou des dirigeants de l'entreprise elle-même. Pour notre cas, l'entreprise « Apple » présente de bons signes financières pour le futur.