Université Iba Der THIAM de Thiès

UFR Sciences et Technologies

Département Mathématique

Master Sciences de Données et Applications option Statistiques et Econométrie

Option : Statistique Econométrie et Modélisation

Projet 1 de Statistiques des Valeurs Extrêmes

Nom des membres du groupe :

Ismael YODA

Amsatou DIOP

Nom du Professeur :

Dr Mouhamad ALLAYA

Année Scolaire 2020 & 2021

Exercice 1

1. Importation des données

```
data=read.csv2("C:/Users/YODA ISMAEL/Desktop/Dossier Etudes/Dossiers Master/Semestre3/Valeurs Extremes/precip_extr.csv",sep = ";", dec=".",header=T)
head(data)
```

```
##   STAID  SOUID     DATE RR Q_RR
## 1    33 100105 19470101  2    0
## 2    33 100105 19470102  2    0
## 3    33 100105 19470103  0    0
## 4    33 100105 19470104  0    0
## 5    33 100105 19470105 60    0
## 6    33 100105 19470106  4    0
```

Nous allons ici supprimer les lignes contenant des valeurs manquantes. Il s'agit ici de toutes les observations de l'année 2017.

```
data <- subset(data, data$Q_RR == 0)[,c("DATE", "RR")]
head(data)
```

```
##       DATE RR
## 1 19470101  2
## 2 19470102  2
## 3 19470103  0
## 4 19470104  0
## 5 19470105 60
## 6 19470106  4
```

Conversion de la variable DATE en format date reconnu par R.

```
data$DATE=as.Date(as.character(data$DATE), format = "%Y%m%d")
head(data)
```

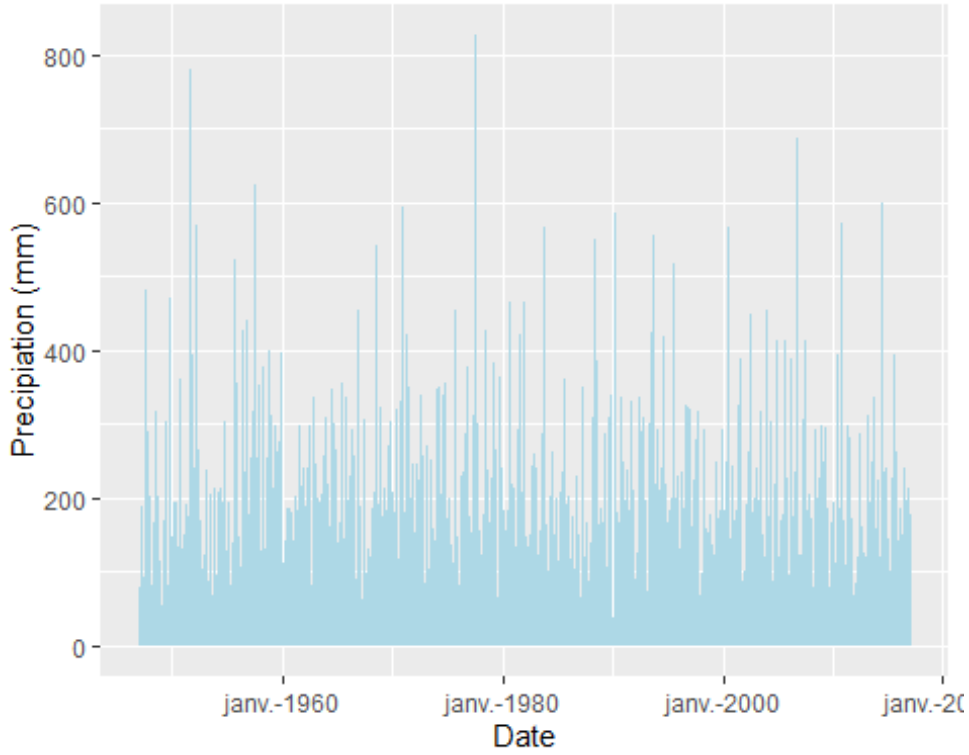
```
##         DATE RR
## 1 1947-01-01  2
## 2 1947-01-02  2
## 3 1947-01-03  0
## 4 1947-01-04  0
## 5 1947-01-05 60
## 6 1947-01-06  4
```

Représentation graphique des données

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
graph <- ggplot(data, aes(x = DATE, ymax = RR, ymin = 0)) +
  geom_linerange(col = "lightblue") +
  scale_x_date(date_labels = "%b-%Y") +
  ylab("Precipiation (mm)") +
  xlab("Date")

graph
```



2. Importation du package evd

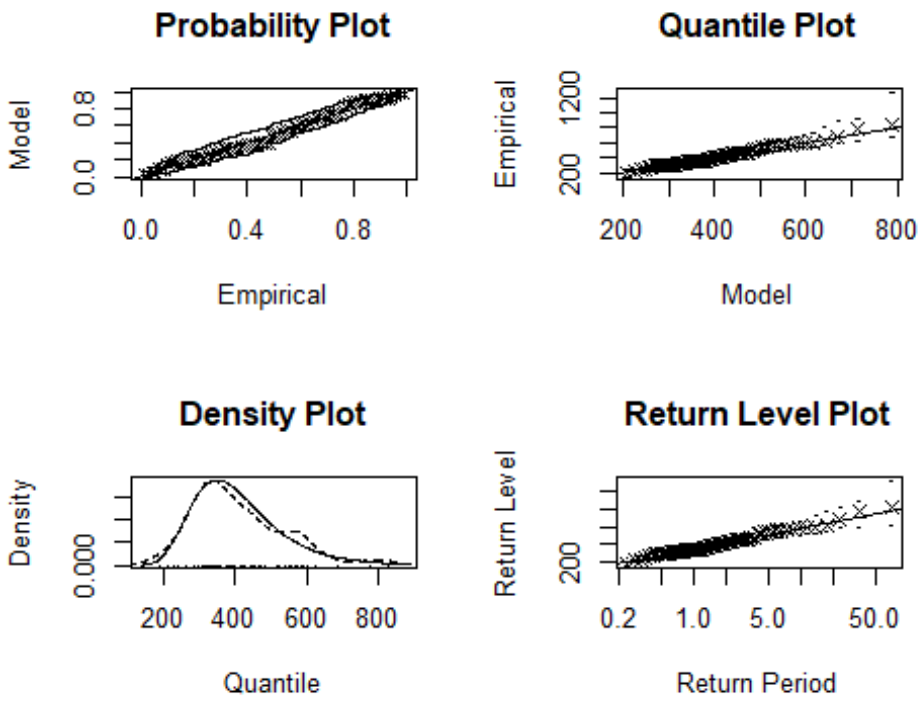
```
library(evd)
```

3. Extraction des maximums par année

```
library(chron)

max_annuel <- aggregate(RR~years(DATE), FUN = max, data = data)
head(max_annuel)

##   years(DATE)  RR
## 1        1947 483
## 2        1948 316
## 3        1949 470
## 4        1950 361
## 5        1951 780
## 6        1952 570
```

4. Estimation d'une GEV avec les maximums annuels en utilisant la fonction fgev avec une réprésentation graphique

```
library(evd)
fitted=fgev(max_annuel$RR)
par(mfrow = c(2, 2))
plot(fitted)
```



Sur la base du tracé Quantile Plot, l'ajustement semble être bon. Nous constatons également que la densité de probabilité a une queue de distribution lourde.

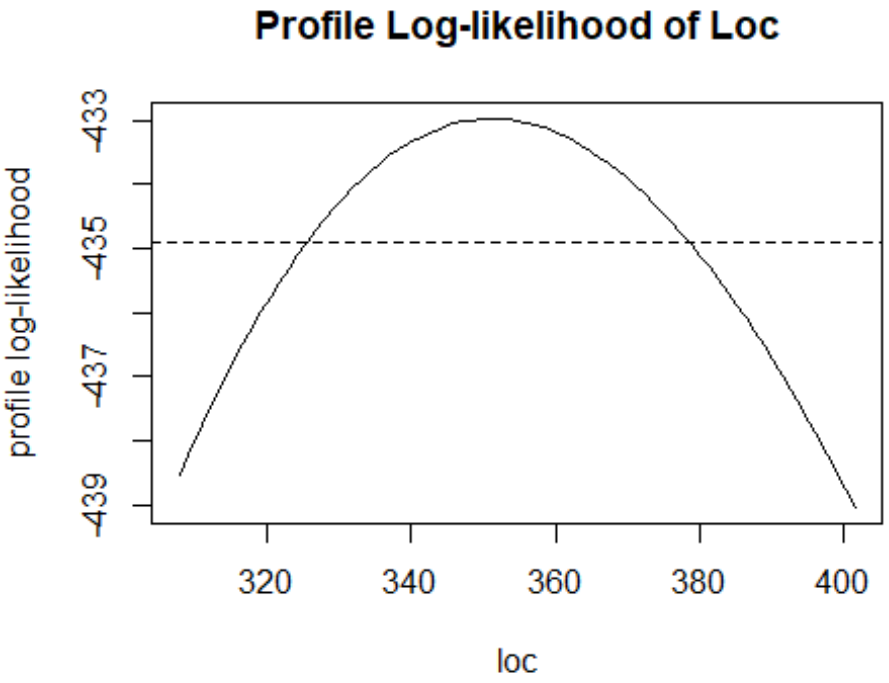
5. Donnons un intervale de confiance des paramètres de la GEV μ, σ et ζ

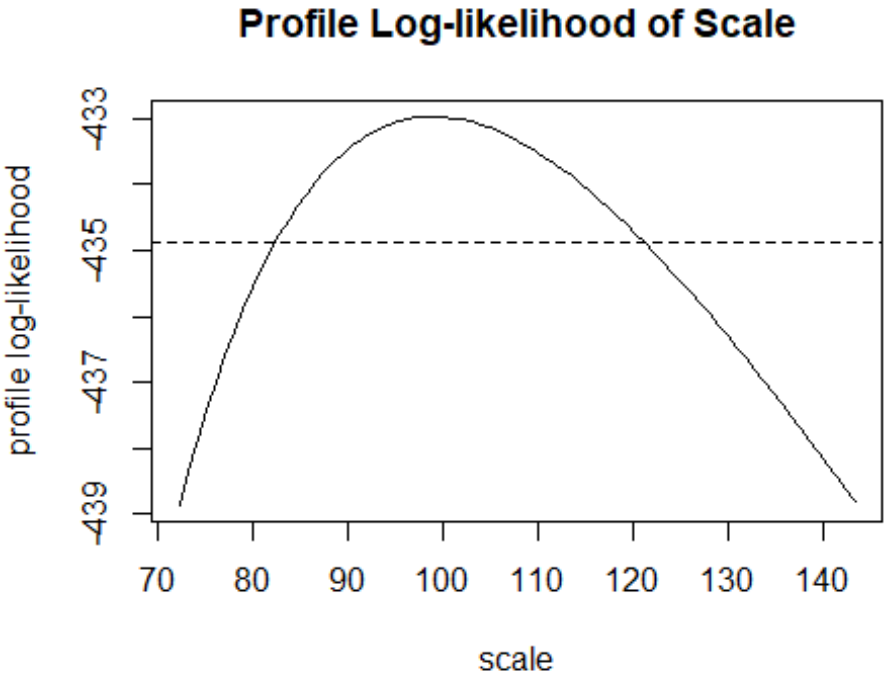
```
# Interval de confiance en utilisant la normalité asymptotique
inter_c <- cbind(low = fitted$par - qnorm(0.975) * fitted$std.err,
          up = fitted$par + qnorm(0.975) * fitted$std.err)
inter_c

##              low         up
## loc   325.1579721 377.5929566
## scale  80.0665881 118.5375099
## shape  -0.1643042   0.1929767
```

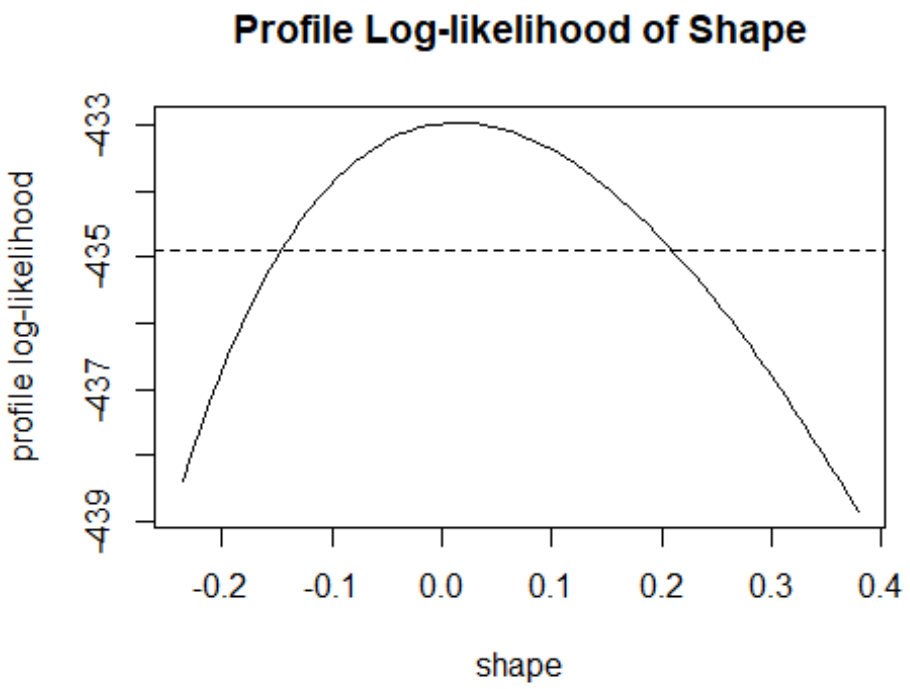
6. Comparons l'intervalle de confiance ci-dessus avec celui obtenue par

```
plot(profile(fitted))
```

```
## [1] "profiling loc"
## [1] "profiling scale"
## [1] "profiling shape"
```


Profile Log-likelihood of Loc
profile log-likelihood
-433
-435
-437
-439
320
340
360
380
400
loc


Profile Log-likelihood of Scale
profile log-likelihood
-433
-435
-437
-439
70
80
90
100
110
120
130
140
scale



La principale différence entre ces deux types d'intervalles de confiance est que celle fournis par "plot(profile(fitted))" est asymétrique. Elle sera donc préferé au premier.

7. Passons l'argument shape=0 à la fonction fgev

```
(fitted1 <- fgev(max_annuel$RR, shape = 0))

##
## Call: fgev(x = max_annuel$RR, shape = 0)
## Deviance: 865.9698
##
## Estimates
##    loc   scale
## 352.17   99.63
##
## Standard Errors
##    loc   scale
## 12.528   9.381
##
## Optimization Information
##   Convergence: successful
##   Function Evaluations: 6
##   Gradient Evaluations: 4
```

Nous pouvons dire qu'il s'agit d'une distribution de Gumbel car cette dernière admet le paramètre ζ=0. Pour vérifier si ce modèle est approprié, nous allons la comparer avec le premier à travers un test anova.

```r
anova(fitted,fitted1)
```

```
## Analysis of Deviance Table
##
##         M.Df Deviance Df  Chisq Pr(>chisq)
## fitted     3   865.94
## fitted1    2   865.97  1 0.0269     0.8697
```

Nous constatons à travers le test anova que le modèle “fitted1” est meilleur que le modèle “fitted” car l’hypothèse nulle selon laquelle le modèle “fitted1” a une plus petite variance que le modèle “fitted” ne peut pas être rejeter.

8. Donnons une estimation des niveaux de retours sur 2 ans, 10 ans et 100 ans

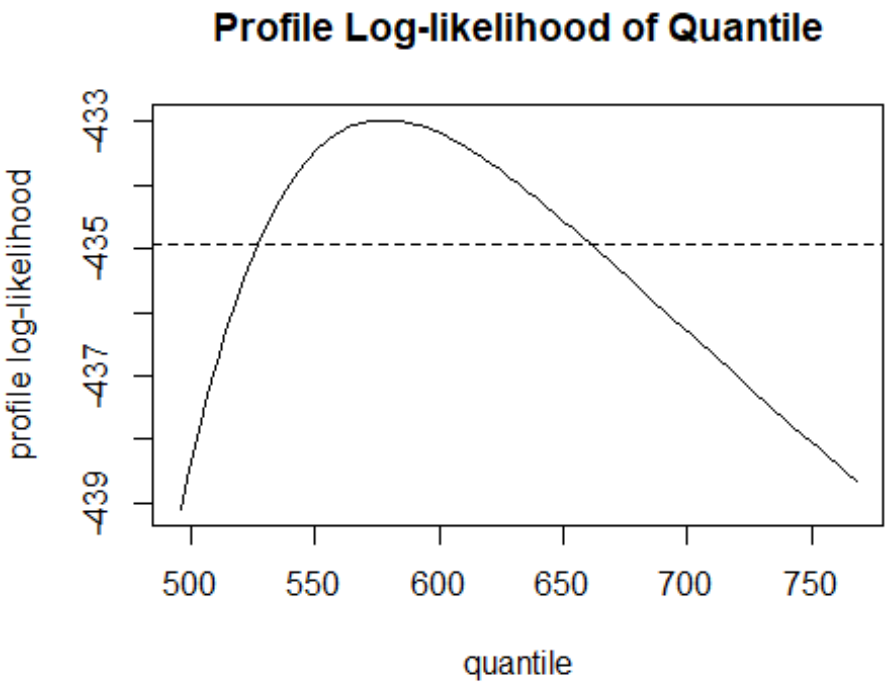
```r
niv_retour <- c(2, 10, 100)
print(qgev(1 - 1/niv_retour, fitted1$par[1], fitted1$par[2], fitted1$par[3]))
```

```
## [1] 388.6856 576.3674 810.4678
```

9. Donnons un intervalle de confiance basé sur la vraissemblance du profil pour le niveau de retour sur 10 ans et commentons

```r
plot(profile(fgev(max_annuel$RR, prob = 1 / 10), "quantile"))
```

```
## [1] "profiling quantile"
```



L'intervalle de confiance pour le niveau de retour est très asymétrique comme il est habituel pour les extrêmes.

10.

11. Essayons d'ajuster un modèle GEV non stationnaire,par exemple, avec une tendance linéaire pour le paramètre de localisation μ

Pour permettre une tendance linéaire en μ dans le temps, ydat devra être une matrice avec juste une seule colonne, où les valeurs de la colonne sont un compteur de temps de 1 à 70 (comme nous avons 70 maxima annuels). Ainsi,

```
ti=matrix(ncol=1,nrow=70)
ti[,1]=seq(1,70,1)
```

Maintenant, pour ajuster le GEV pour permettre une tendance linéaire en μ, nous tapons:

```
library(ismev)

fitted2=gev.fit(max_annuel$RR,ydat=ti,mul=1)

## $model
## $model[[1]]
```

```
## [1] 1
##
## $model[[2]]
## NULL
##
## $model[[3]]
## NULL
##
##
## $link
## [1] "c(identity, identity, identity)"
##
## $conv
## [1] 0
##
## $nllh
## [1] 432.9695
##
## $mle
## [1] 349.90437420   0.03434522  98.98185255   0.01632223
##
## $se
## [1] 25.81325561  0.58149081  9.85917378  0.09392654
```

Exercice 2

Dans cet exercice, nous effectuerons une analyse des valeurs extrêmes en utilisant les dépassements de seuil pour Yahoo log-retours.

1. Installons et chargeons d'abord le quantmod et obtenons les prix quotidiens de apple en appelant getSymbols ("AAPL").

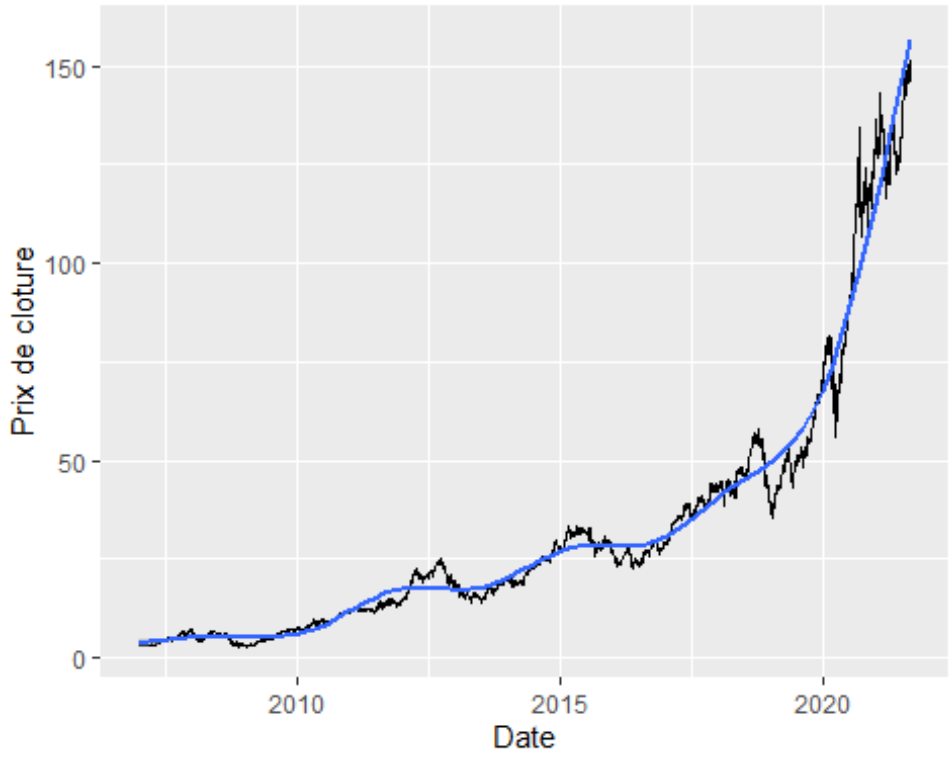
```
library(quantmod)

library(tidyquant)

getSymbols("AAPL",src="yahoo")

## [1] "AAPL"

head(AAPL)

##            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
## 2007-01-03  3.081786  3.092143 2.925000   2.992857  1238319600      2.569716
## 2007-01-04  3.001786  3.069643 2.993571   3.059286   847260400      2.626753
## 2007-01-05  3.063214  3.078571 3.014286   3.037500   834741600      2.608047
## 2007-01-08  3.070000  3.090357 3.045714   3.052500   797106800      2.620926
## 2007-01-09  3.087500  3.320714 3.041071   3.306071  3349298400      2.838647
## 2007-01-10  3.383929  3.492857 3.337500   3.464286  2952880000      2.974493
```

2. Tracons la série chronologique brute ainsi que les rendements logarithmiques négatifs (en utilisant les prix de clôture).

```
cloture <- data.frame(Date = index(AAPL),
                      Close = AAPL$AAPL.Close)

rend_log <- data.frame(Date = index(AAPL),
                       rend_log = -diff(log(AAPL$AAPL.Close)))
```

Tracé de la série brute à l'aide de ggplot

```
library(ggplot2)
ggplot(cloture, aes(x = Date, y = AAPL.Close)) +
  geom_line() +
  ylab("Prix de cloture") +
  geom_smooth()
```



La série des cours de clôture montre une tendance linéaire. Elle n'est donc pas stationnaire et la théorie des valeurs extrêmes standard qui suppose une série stationnaire, ne pourrait pas marcher avec elle. Nous allons effectuer un test de stationnarité pour appuyer nos analyses.

Effectuons pour ce faire, le test de stationnarité de Dickey Fuller. L'hypothèse nulle du test est que la série n'est pas stationnaire.

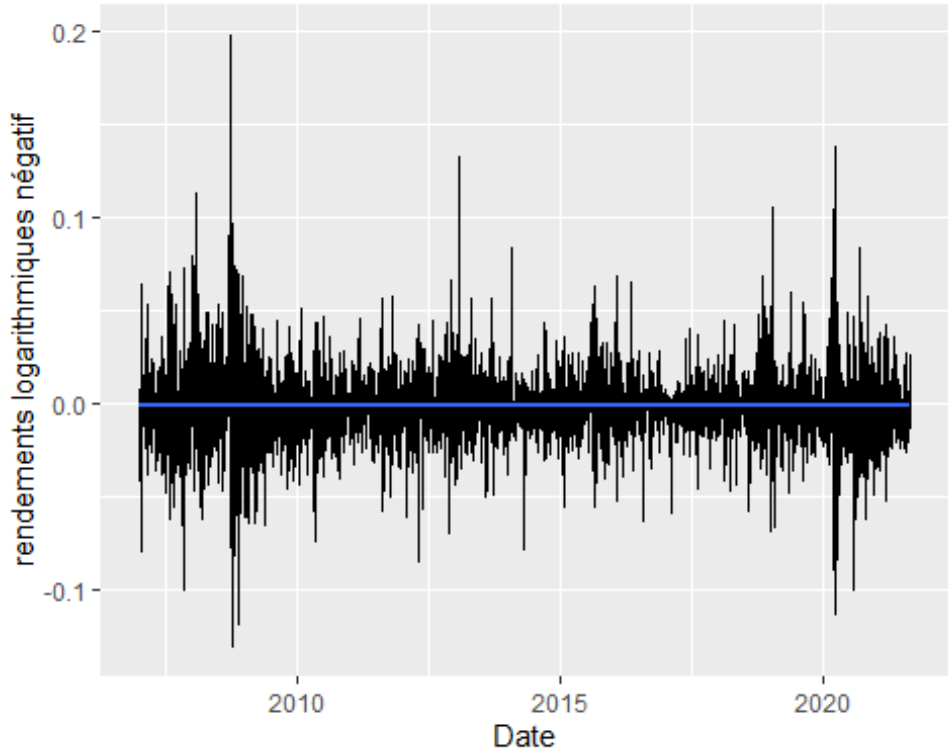
```
library(tseries)
```

```
adf.test(AAPL$AAPL.Close)

## Warning in adf.test(AAPL$AAPL.Close): p-value greater than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  AAPL$AAPL.Close
## Dickey-Fuller = 1.147, Lag order = 15, p-value = 0.99
## alternative hypothesis: stationary
```

D'après les résultats du test la p-value est supérieure à tous les seuils conventionnels .Notre série n'est donc pas stationnaire.

Tracé de la série des rendements logarithmiques négatifs

```
ggplot(rend_log, aes(x = Date, y = AAPL.Close)) +
  geom_line() +
  ylab("rendements logarithmiques négatif") +
  geom_smooth()
```



En observant la série des rendements logarithmiques négatifs, nous constatons que la moyenne et la variance semble être constantes dans le temps. Nous pouvons dire que cette série est stationnaire.

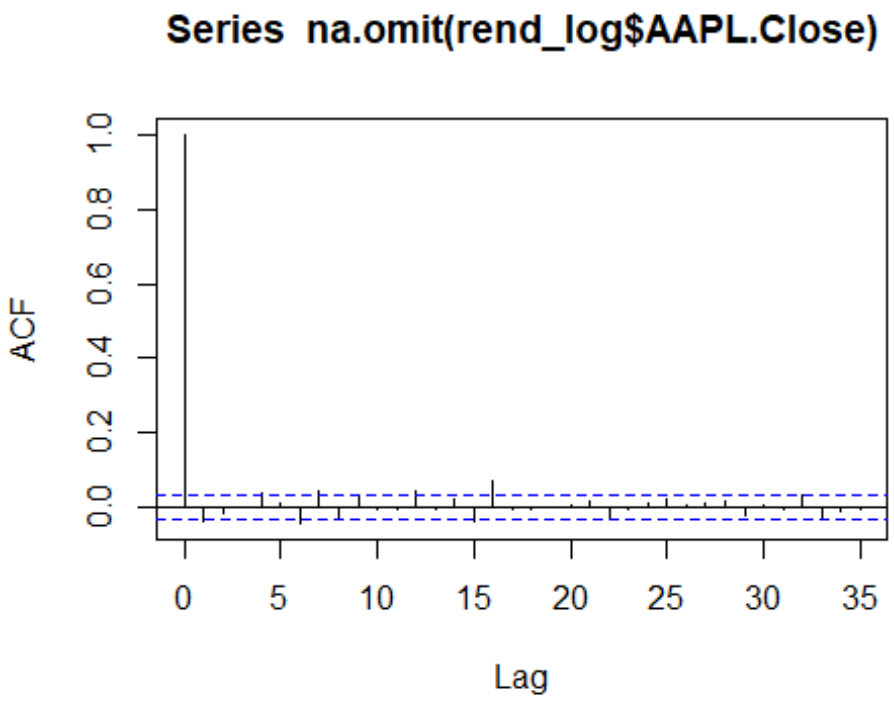
Effectuons à nouveau le test de stationnarité de Dickey Fuller pour vérifier la stationnarité de la série des rendements logarithmiques négatifs.

```r
library(tseries)
adf.test(na.omit(rend_log$AAPL.Close))
```

```
## Warning in adf.test(na.omit(rend_log$AAPL.Close)): p-value smaller than printed
## p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  na.omit(rend_log$AAPL.Close)
## Dickey-Fuller = -13.683, Lag order = 15, p-value = 0.01
## alternative hypothesis: stationary
```

D'aprés les résultats du test, la p-value est inférieure à tous les seuils conventionnels .Notre série des rendements logarithmiques négatifs est donc stationnaire. C'est cette série qui sera utilisé pour estimer notre modèle. Nous allons par ailleurs vérifier s'il n'existe pas une autocorrélation de notre série en traçant l'autocorrélogramme partiel(ACF).

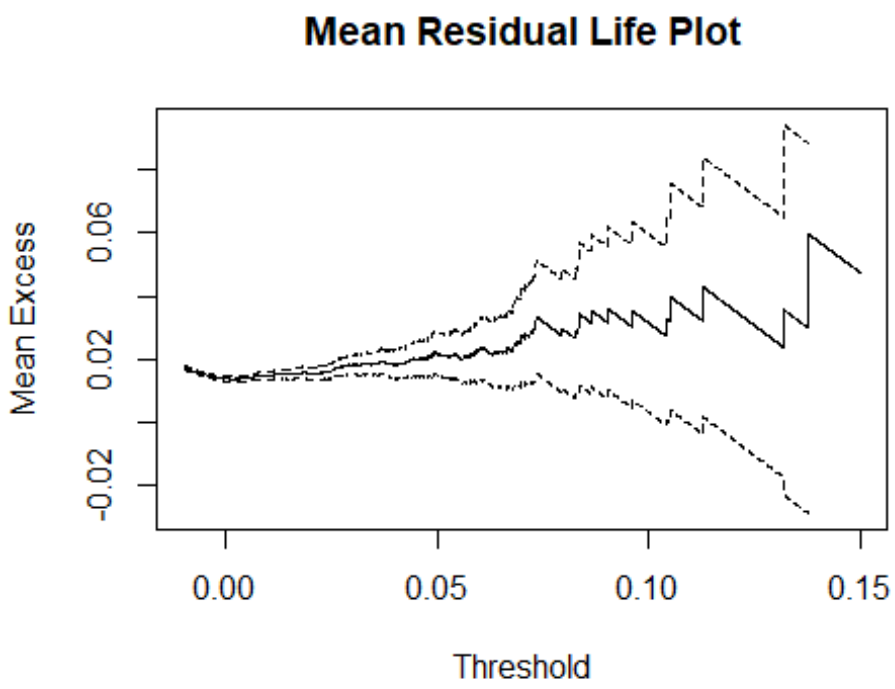
```r
acf(na.omit(rend_log$AAPL.Close))
```



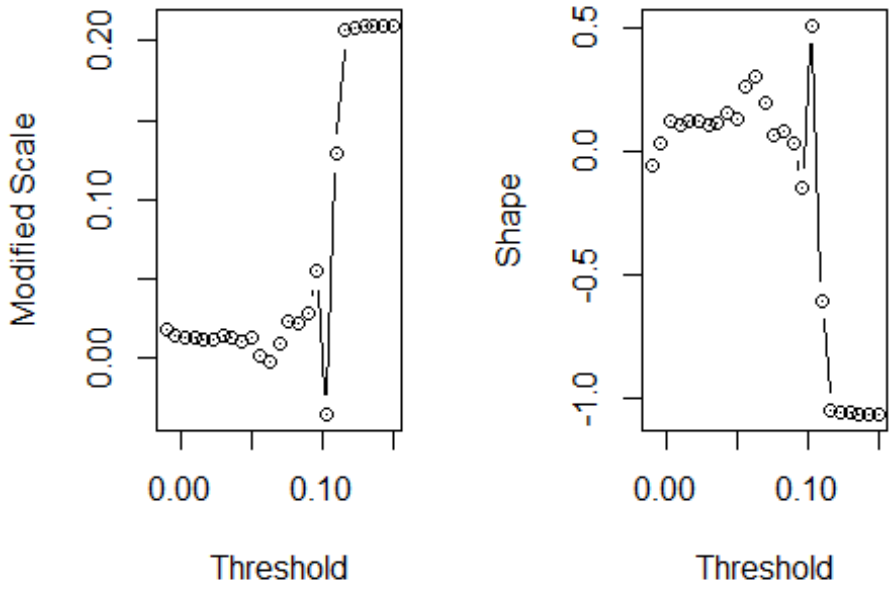
La majorité des pics n'étant pas significatifs, alors nous ne pouvons pas soupçonner la présence d'une autocorrélation.

3. À l'aide des fonctions mrlplot et tcpot, identifions les valeurs de seuils sensibles afin que les dépassements puisse raisonnablement être considéré comme GPD

```
mrlplot(rend_log$AAPL.Close,c(-0.01,0.15))
```




```
par(mfrow = c(1, 2))
tcplot(rend_log$AAPL.Close, c(-0.01, 0.15),std.err=F)
```



Sur la base des tracés précédents, un seuil autour de 0,05 devrait être bon.

```
thresh=0.05
```

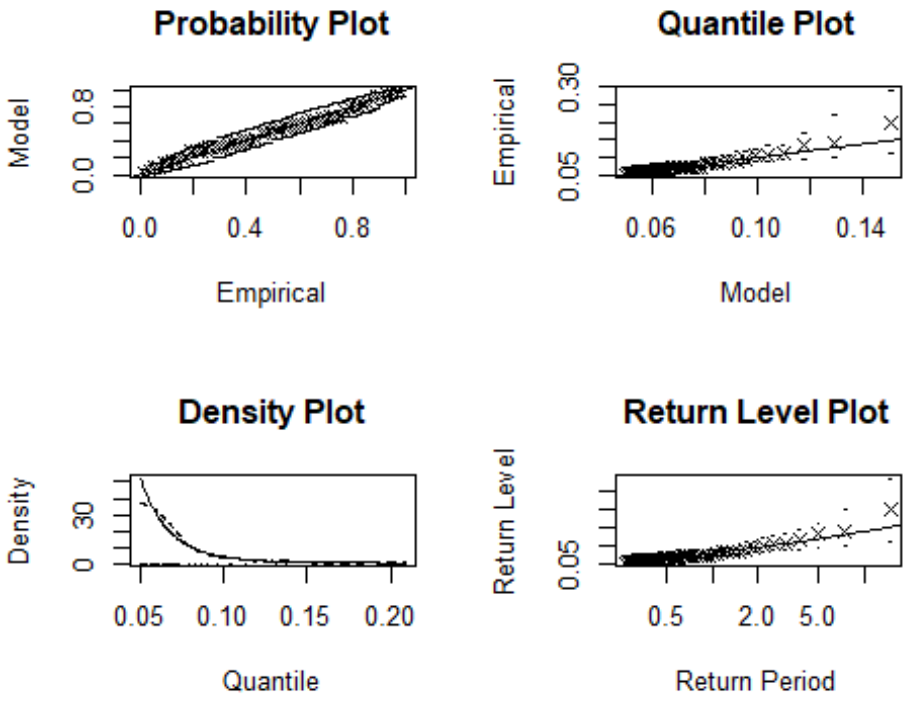
4.Suivons les mêmes étapes que dans l'exercice précédent

Estimation d'un GPD en considérant le seuil maximal=0,05

```
(fit <- fpot(rend_log$AAPL.Close, thresh, npp = 250))

#npp' signifie "Nombre d'observations par période".Ici, nous  mettons 'npp = 250' car il y a (environ) 250 jours ouvrables dans une année

##
## Call: fpot(x = rend_log$AAPL.Close, threshold = thresh, npp = 250)
## Deviance: -318.6551
##
## Threshold: 0.05
## Number Above: 56
## Proportion Above: 0.0152
##
## Estimates
##   scale    shape
## 0.01872  0.13410
##
## Standard Errors
##    scale     shape
## 0.003595  0.140641
##
## Optimization Information
##   Convergence: successful
##   Function Evaluations: 38
##   Gradient Evaluations: 6
```

Réprésentation graphique de l'estimation

```
par(mfrow = c(2,2))
plot(fit)
```



Selon le Quantile plot,le modèle semble être bien ajusté.

Interval de confiance à 95% en utilisant la normalité asymptotique

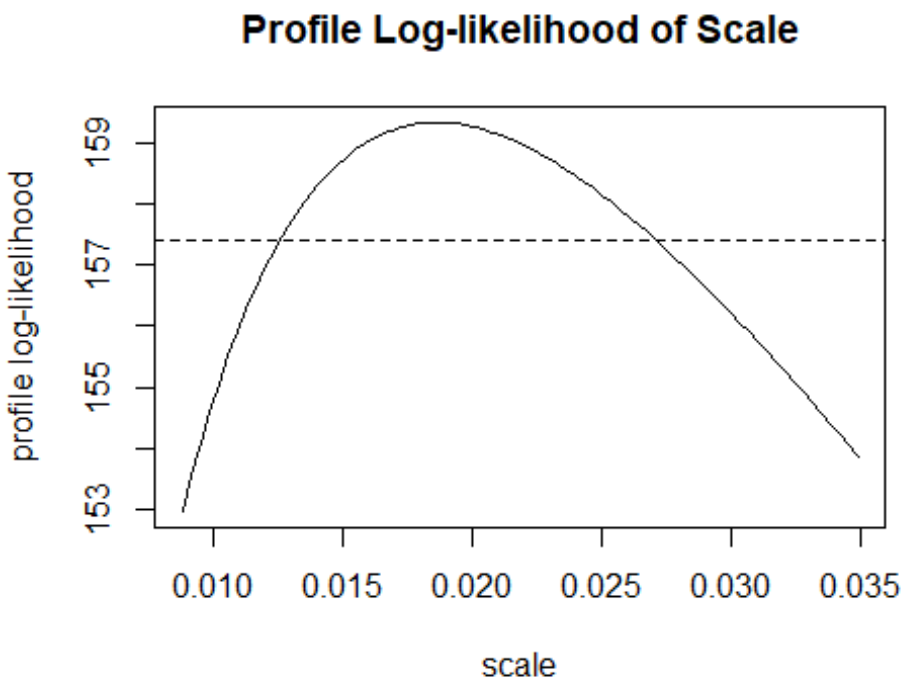
```
interv <- cbind(low = fit$par - qnorm(0.975) * fit$std.err,
          up = fit$par + qnorm(0.975) * fit$std.err)
interv
```

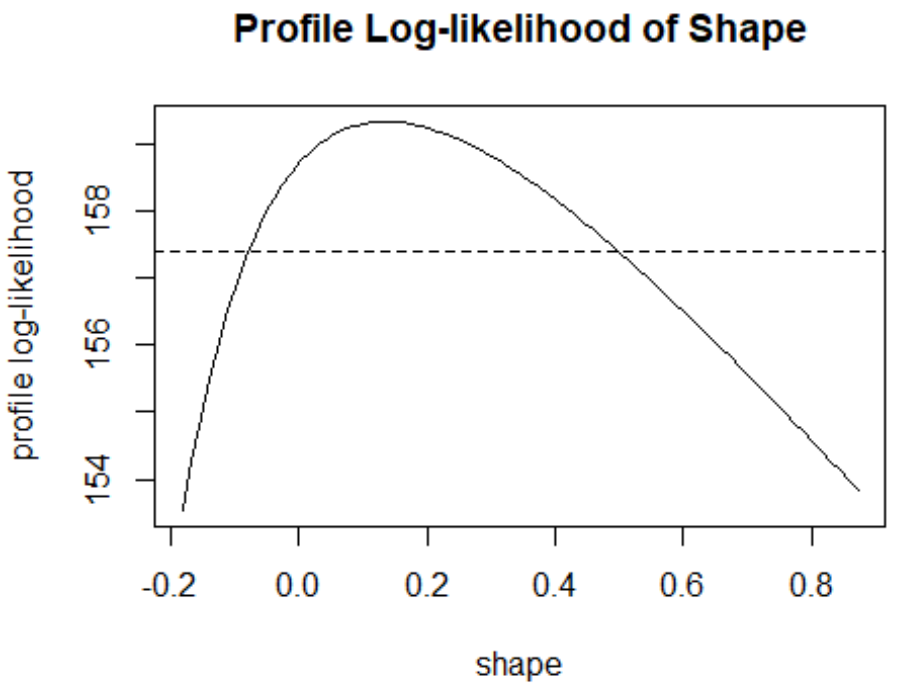
```
##               low        up
## scale  0.01167052 0.02576148
## shape -0.14154697 0.40975679
```

Intervalles de confiance à 95 % basés sur la vraisemblance profilée

```
plot(profile(fit))
```

```
## [1] "profiling scale"
## [1] "profiling shape"
```





Nous choisirons l'interval de confiance basé sur la vraisemblance profilée à cause de sa forte asymétrie.

Estimation d'un second modèle en considérant $\zeta = 0$ (modèle exponentiel)

```
fit1 <- fpot(rend_log$AAPL.Close, thresh, shape = 0, npp = 250)
fit1

##
## Call: fpot(x = rend_log$AAPL.Close, threshold = thresh, npp = 250,      shape = 0)
## Deviance: -317.4154
##
## Threshold: 0.05
## Number Above: 56
## Proportion Above: 0.0152
##
## Estimates
##   scale
## 0.02162
##
## Standard Errors
##    scale
## 0.002871
##
## Optimization Information
##   Convergence: successful
##   Function Evaluations: 19
##   Gradient Evaluations: 1
```

Comparaison des deux modèles avec le test d'anova

```
anova(fit, fit1)

## Analysis of Deviance Table
##
##      M.Df Deviance Df  Chisq Pr(>chisq)
## fit     2  -318.66
## fit1    1  -317.42  1 1.2397     0.2655
```

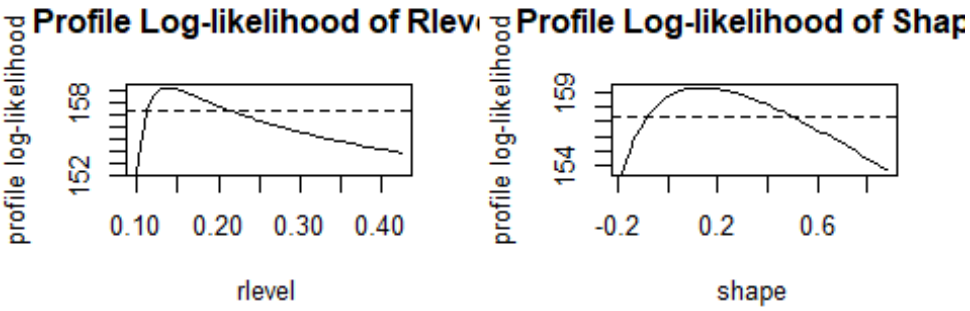
Nous constatons à travers le test anova que le modèle "fit1" est meilleur que le modèle "fit" car l'hypothèse nulle selon laquelle le modèle "fit1" a une plus petite variance que le modèle "fit" ne peut pas être rejeter.

Estimation des niveaux de retour sur 2 ans, 10 ans et 100 ans

```
niv_ret <- c(2, 10, 100)
ret <- mean(rend_log$AAPL.Close > thresh, na.rm = TRUE)
qgpd(1 - 1 / (ret * 250 * niv_ret), thresh, fit1$par["scale"], fit1$par["shape"])

## [1] 0.09384701 0.12864550 0.17843089
```

5. Commentons ce qui se passe lorsque vous passez l'option mper = 10 à la fonction fpot

```
fit3=fpot(rend_log$AAPL.Close, thresh, mper = 10, npp = 250)
fit3

##
## Call: fpot(x = rend_log$AAPL.Close, threshold = thresh, npp = 250,      mper = 10)
## Deviance: -318.6553
##
## Threshold: 0.05
## Number Above: 56
## Proportion Above: 0.0152
##
## Estimates
## rlevel   shape
## 0.1378  0.1360
##
## Standard Errors
##  rlevel    shape
## 0.01905  0.14128
##
## Optimization Information
##   Convergence: successful
##   Function Evaluations: 36
##   Gradient Evaluations: 5

par(mfrow=c(2,2))
plot(profile(fit3))

## [1] "profiling rlevel"
## [1] "profiling shape"
```



En passant l'option mper = 10 à la fonction fpot, nous constatons que la fonction estime uniquement la paramètre shape et le niveau de retour. Le paramètre scale semble avoir pris la valeur 0 par défaut.

Nous pouvons donc conclure que lorsque l'option mper= m est une valeur positive, alors le modèle de Pareto généralisé est reparamétré de manière à ce que les paramètres soient rlevel et shape, où rlevel est le niveau de retour m et shape le paramètre ζ du modèle.

Exercice 3

Jetons un œil à https://www.ecad.eu et choisissons notre ensemble de données (environnementales) préféré. Effectuons une analyse des données extrêmes.

Nous allons étudier les températures maximum extrêmes de la ville de Clermont Ferrand en France. Nous disposons d'une base de données de nos observations qui s'étend de 2020/10/01 à 2021/05/31. Nous n'avons pas pu trouver une base de données plus longue. Nous allons essayer de déterminer la distribution des valeurs maximales mensuelles et effectuer une GEV.

Importation des donnees

```
df=read.csv2("C:/Users/YODA ISMAEL/Desktop/Dossier Etudes/Dossiers Master/Semestre3/Valeurs Extremes/temp_max.csv",sep = ",", dec=".",header=T)
head(df)

##   STAID  SOUID    DATE  TX Q_TX
## 1 21787 218275 20201001 165    0
```

```
## 2 21787 218275 20201002 150    0
## 3 21787 218275 20201003 145    0
## 4 21787 218275 20201004 150    0
## 5 21787 218275 20201005 154    0
## 6 21787 218275 20201006 167    0
```

Transformation de la variable DATE en format date reconnu par R

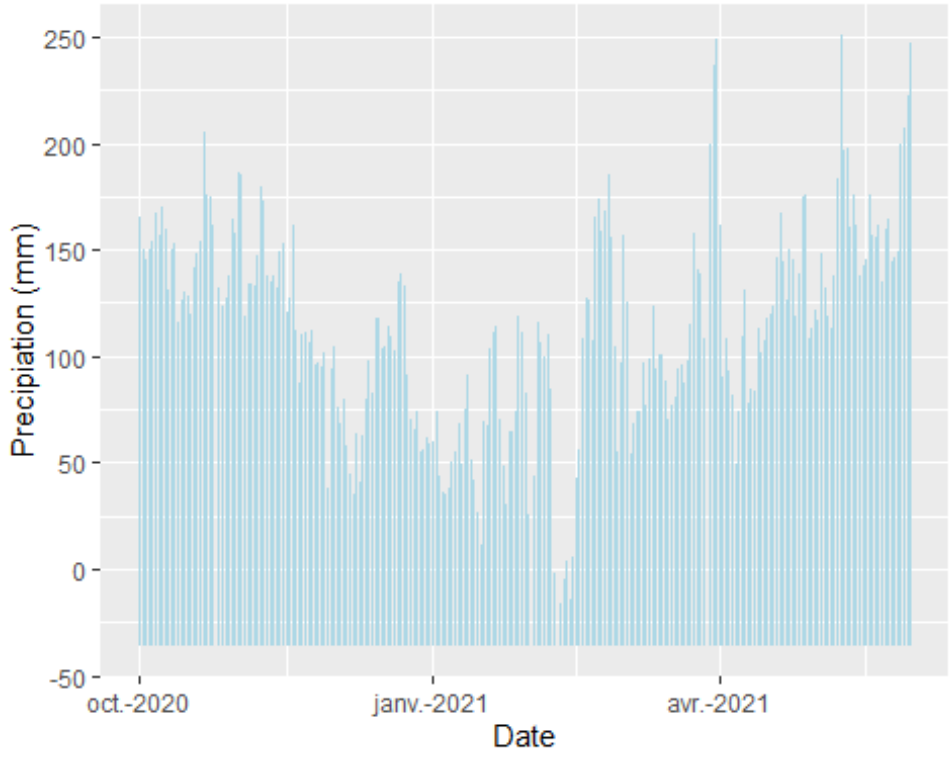
```
df$DATE=as.Date(as.character(df$DATE), format = "%Y%m%d")
```

Suppréssion des valeurs manquantes

```
df <- subset(df, df$Q_TX == 0)[,c("DATE", "TX")]
head(df)

##        DATE  TX
## 1 2020-10-01 165
## 2 2020-10-02 150
## 3 2020-10-03 145
## 4 2020-10-04 150
## 5 2020-10-05 154
## 6 2020-10-06 167
```

Représentation graphique des températures maximums journalières

```
library(ggplot2)
graph <- ggplot(df, aes(x = DATE, ymax = TX, ymin = -36)) +
  geom_linerange(col = "lightblue") +
  scale_x_date(date_labels = "%b-%Y") +
  ylab("Precipiation (mm)") +
  xlab("Date")

graph
```



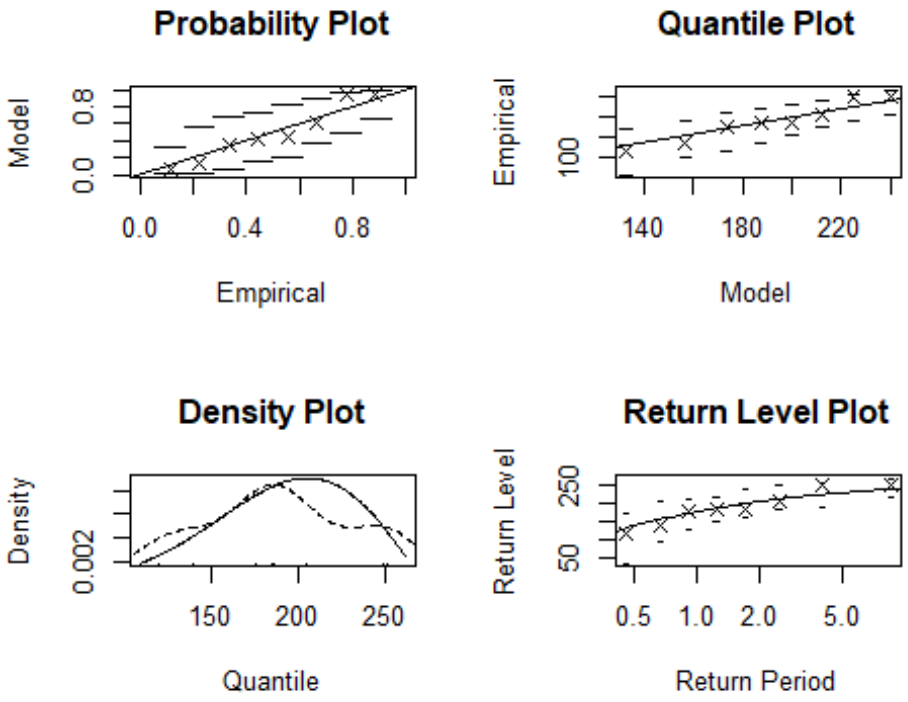
Déterminons la distribution des maximums mensuelles

```
library(chron) ## pour obtenir la fonction years
max_mois <- aggregate(TX~months(DATE), FUN = max, data = df)
max_mois
```

```
##   months(DATE)  TX
## 1        avril  176
## 2     décembre  139
## 3      février  185
## 4      janvier  119
## 5          mai  251
## 6         mars  249
## 7     novembre  186
## 8      octobre  205
```

Estimation et représentation graphique du modèle

```
library(evd)
fittede=fgev(max_mois$TX)
par(mfrow = c(2, 2))
plot(fittede)
```



Sur la base du tracé Quantile plot, l'ajustement semble être bon.

Les valeurs des estimateurs des paramètres

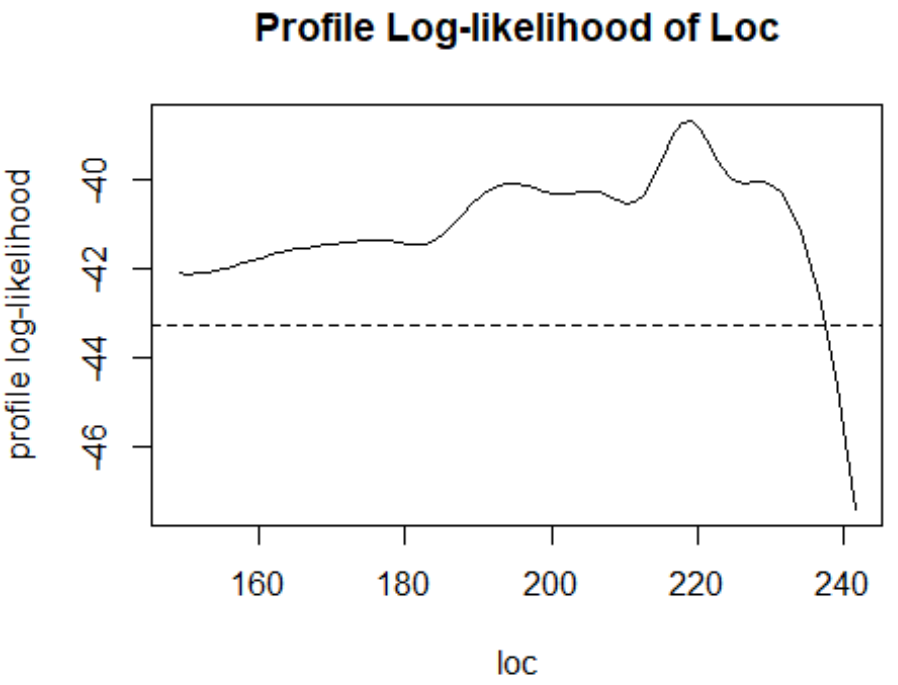
```
fittede$param

##         loc       scale       shape
## 177.9491556  47.1128390  -0.4941698
```

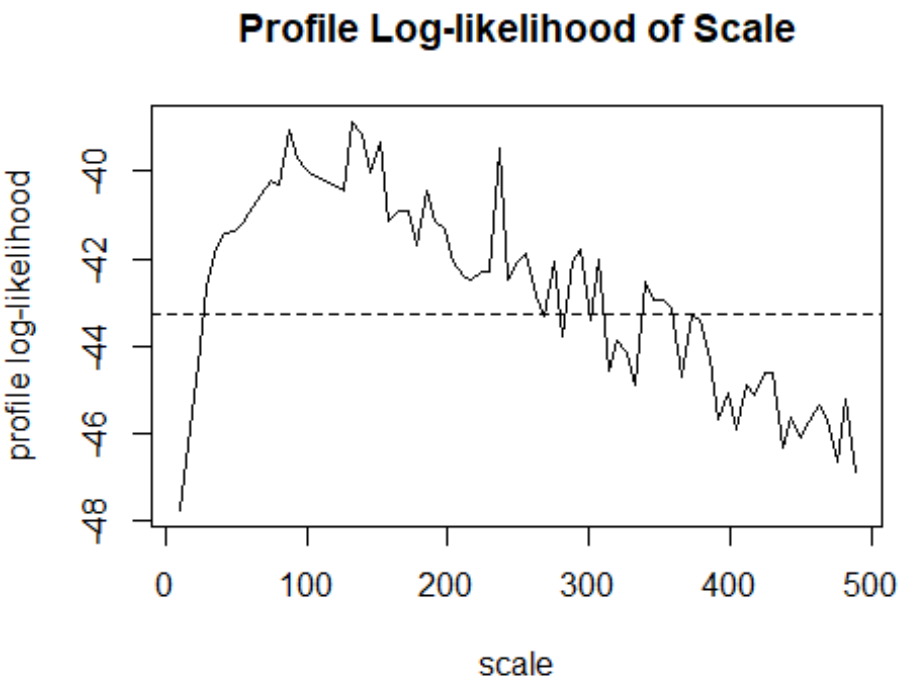
Nous constatons que le pramètre ζ est inferieur à zéro ce qui est la caractéristique d'une distribution de Weibull.

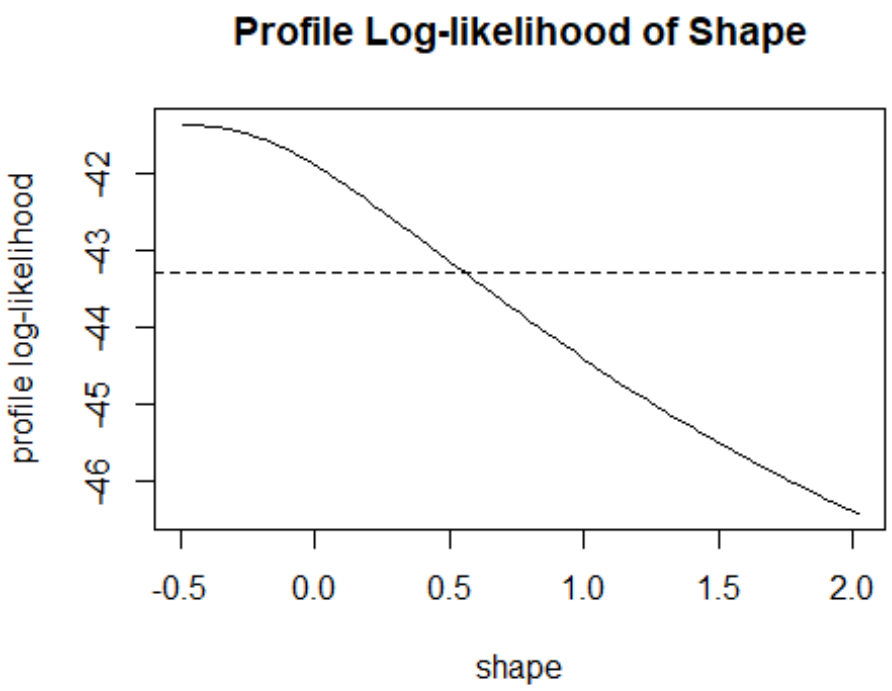
Donnons un interval de confiance des paramètres de la GEV μ, σ et ζ

```
# Interval de confiance en utilisant la normalité asymptotique
inter_c <- cbind(low = fittede$par - qnorm(0.975) * fittede$std.err,
          up = fittede$par + qnorm(0.975) * fittede$std.err)
inter_c

##              low        up
## loc   132.617981 223.280330
## scale   5.443115  88.782563
## shape  -1.808958   0.820618

#l'intervalle de confiance basé sur la vraisemblance profilée
plot(profile(fittede))
```

```
## [1] "profiling loc"

## Warning in profile.evd(fittede): If loc is to satisfy `conf', `mesh' must be
## smaller

## [1] "profiling scale"
## [1] "profiling shape"

## Warning in profile.evd(fittede): If shape is to satisfy `conf', `mesh' must be
## smaller

## Warning in profile.evd(fittede): If shape is to satisfy `conf', `mesh' must be
## smaller
```


Profile Log-likelihood of Loc
profile log-likelihood
-40
-42
-44
-46
160
180
200
220
240
loc


Profile Log-likelihood of Scale
profile log-likelihood
-40
-42
-44
-46
-48
0
100
200
300
400
500
scale



Nous pouvons maintenir les mêmes conclusions que dans nos analyses précédentes sur le fait que cet intervalle de confiance fournis de meilleures résultats que celle utilisant la normalité asymptotique et cela, dû au fait de son asymétrie.

Considérons que notre distribution des extrêmes est celle de Gumbel c'est à dire avec $\zeta = 0$.

```
(fittede1 <- fgev(max_mois$TX, shape = 0))

##
## Call: fgev(x = max_mois$TX, shape = 0)
## Deviance: 83.78117
##
## Estimates
##   loc  scale
## 167.0   40.2
##
## Standard Errors
##   loc  scale
## 15.06  10.85
##
## Optimization Information
##   Convergence: successful
##   Function Evaluations: 10
##   Gradient Evaluations: 8
```

Comparons les deux modèles à travers un test anova

```
anova(fittede, fittede1)
```

```
## Analysis of Deviance Table
##
##          M.Df Deviance Df  Chisq Pr(>chisq)
## fittede     3   82.724
## fittede1    2   83.781  1 1.0567      0.304
```

D'apres le test anova, on ne peut pas rejetter l'hypothèse nulle selon laquelle le modèle "fittede1" a une plus petite variance que le modele "fittede". Le modèle "fittede1" est donc meilleure que "fittede".

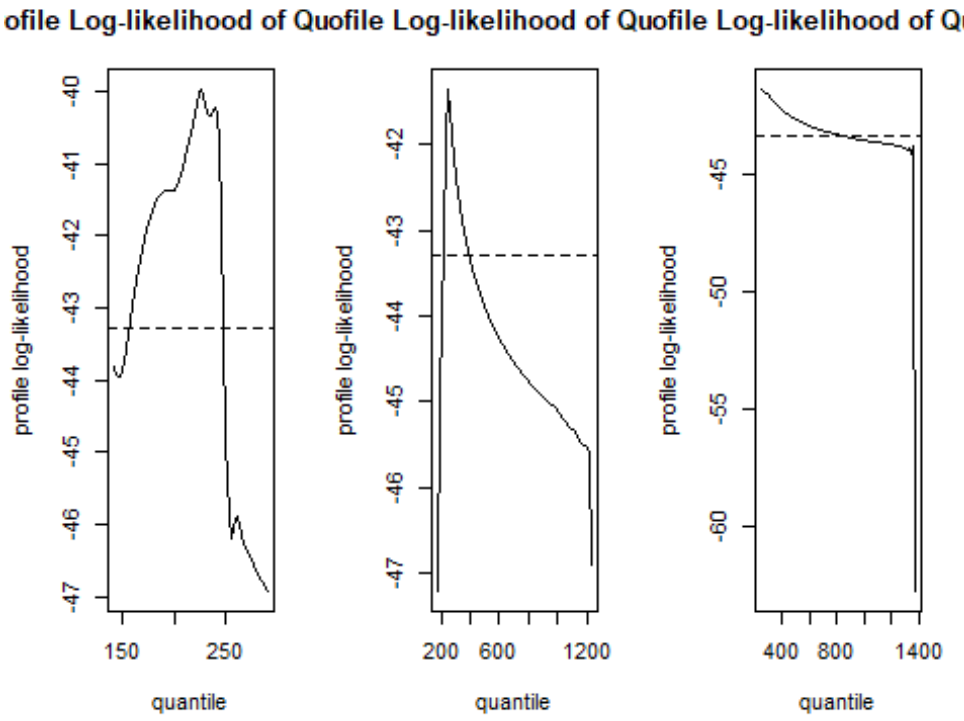
Estimation du niveau de retour pour 2, 10 et 50 ans avec le meilleur modèle.

```
niv_retour <- c(2, 10, 50)
print(qgev(1 - 1/niv_retour, fittede1$par[1],fittede1$par[2], fittede1$par[3]))
```

```
## [1] 181.6969 257.4350 323.8344
```

Donnons un intervalle de confiance de la prévision. Nous utiliserons l'intervalle de confiance basé sur la vraisemblance profilée car il fourni de meilleures résultats.

```
par(mfrow=c(1,3))
plot(profile(fgev(max_mois$TX, prob = 1 / 2), "quantile"))

plot(profile(fgev(max_mois$TX, prob = 1 / 10), "quantile"))

plot(profile(fgev(max_mois$TX, prob = 1 / 50), "quantile"))
```



Nous allons étudier à présent une base de donnée financière (QQQ) issue de la Google finance. Les actions de QQQ comprennent 100 des plus grandes sociétés du Nasdaq, telles qu'Apple, Amazon, Google et Facebook. Nous disposons d'une base de données de nos observation qui s'étendent de 2007-01-03 à ce jour. Nous allons essayer de déterminer la distribution des valeurs maximales annuelle et effectuer une GPD.

Importation des données

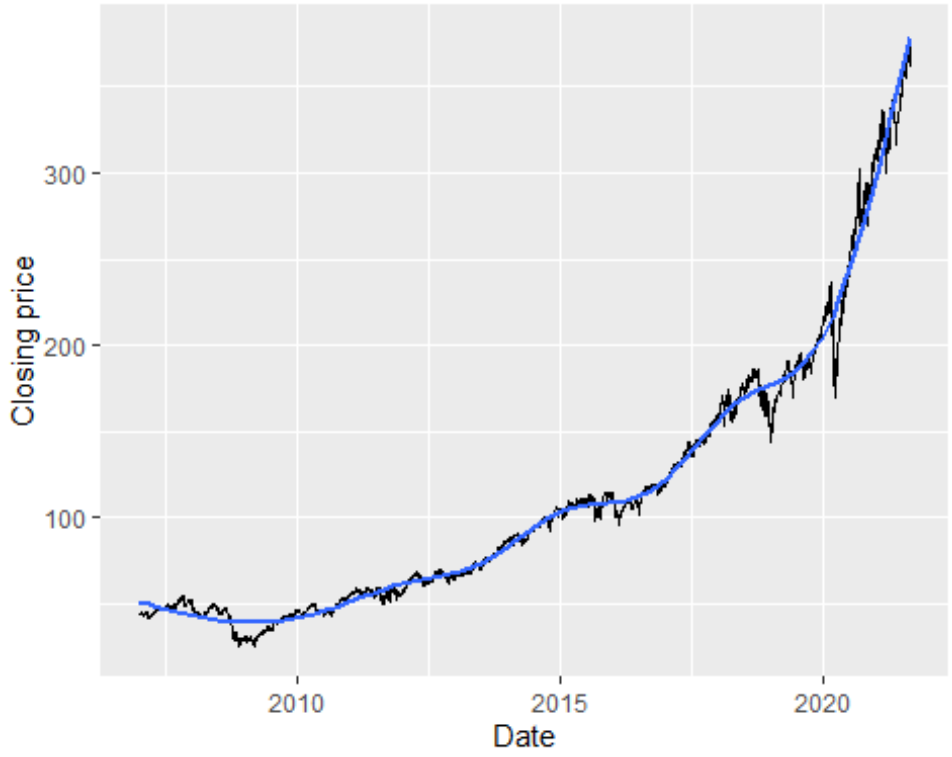
```
library(quantmod)
getSymbols(Symbols = "QQQ", auto_assign = TRUE)

## [1] "QQQ"

head(QQQ)

##            QQQ.Open QQQ.High QQQ.Low QQQ.Close QQQ.Volume QQQ.Adjusted
## 2007-01-03    43.46    44.06   42.52     43.24  167689500     38.16515
## 2007-01-04    43.30    44.21   43.15     44.06  136853500     38.88892
## 2007-01-05    43.95    43.95   43.48     43.85  138958800     38.70357
## 2007-01-08    43.89    44.12   43.64     43.88  106401600     38.73004
## 2007-01-09    44.01    44.29   43.63     44.10  121577500     38.92422
## 2007-01-10    43.96    44.66   43.82     44.62  121070100     39.38319
```

Séries chronologique des niveaux des prix de clotures et celle des rendements logarithmiques négatifs.

```
cloture <- data.frame(Date = index(QQQ),
                      Close = QQQ$QQQ.Close)
rend_negatif <- data.frame(Date = index(QQQ),
                       rend_negatif = -diff(log(QQQ$QQQ.Close)))
```

Représentation graphiques de la série chronologique des prix à la cloture avec ggplot2

```
ggplot(cloture, aes(x = Date, y = QQQ.Close)) +
  geom_line() +
  ylab("Closing price") +
  geom_smooth()
```



Au vu du tracé de la série, nous constatons la présence d'une tendance croissante. Nous pouvons dire qu'elle n'est pas stationnaire. Nous allons vérifier la stationnarité avec le test de Dickey Fuller. L'hypothèse nulle du test que la série n'est pas stationnaire.

```
library(tseries)
adf.test(na.omit(QQQ$QQQ.Close))

## Warning in adf.test(na.omit(QQQ$QQQ.Close)): p-value greater than printed p-
## value

##
##  Augmented Dickey-Fuller Test
##
## data:  na.omit(QQQ$QQQ.Close)
## Dickey-Fuller = 1.4069, Lag order = 15, p-value = 0.99
## alternative hypothesis: stationary
```

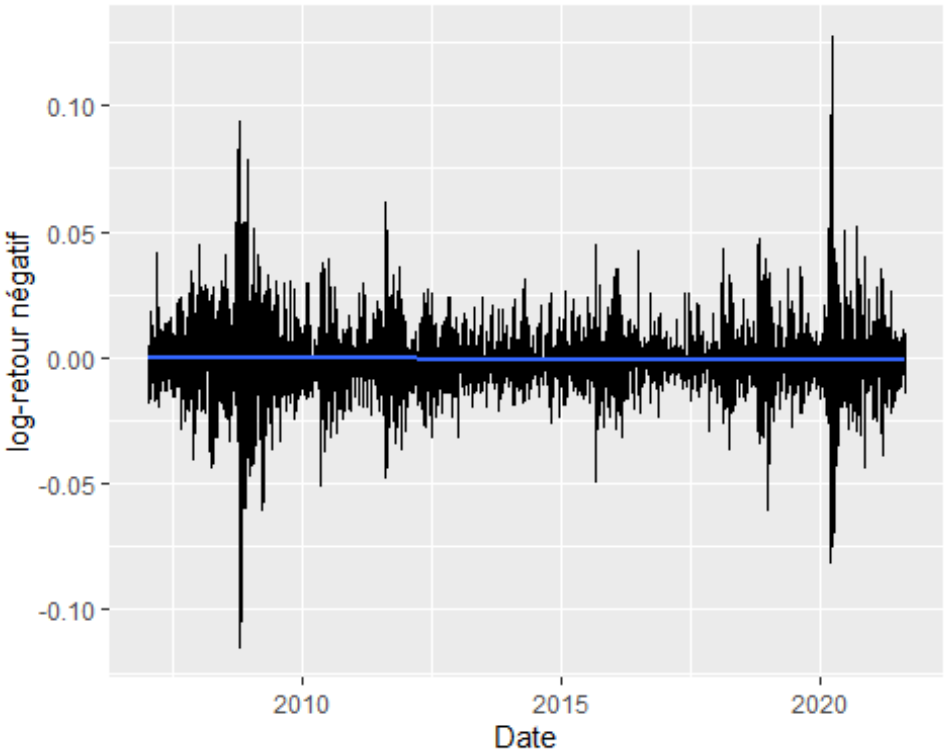
Les résultats du test ci-dessus montrent que la p-value est supérieure à tous les seuils conventionnels. La série brute des prix à la clôture n'est donc pas stationnaire.

Les méthodes standards de la théorie des valeurs extrêmes qui suppose que la série est stationnaire, ne peut donc pas marcher dans ce cas de figure. Nous allons étudier la stationnarité de la série des rendements logarithmiques négatifs.

Représentation graphique de la série chronologique des rendements logarithmiques négatifs.

```
library(ggplot2)
ggplot(rend_negatif, aes(x = Date, y = QQQ.Close)) +
```

```
geom_line() +
ylab("log-retour négatif") +
geom_smooth()
```

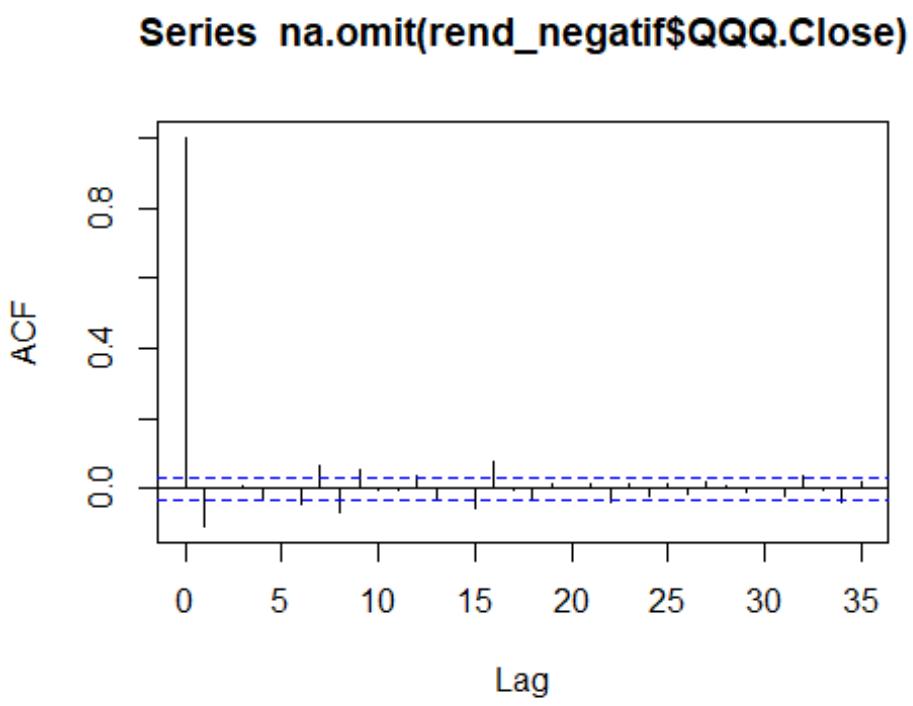


En observant la série, nous constatons que la moyenne et la variance semblent être constantes dans le temps. La série semble stationnaire. Nous allons vérifier avec un nouveau test des Dickey Fuller.

```
library(tseries)
adf.test(na.omit(rend_negatif$QQQ.Close))

## Warning in adf.test(na.omit(rend_negatif$QQQ.Close)): p-value smaller than
## printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  na.omit(rend_negatif$QQQ.Close)
## Dickey-Fuller = -15.089, Lag order = 15, p-value = 0.01
## alternative hypothesis: stationary
```

D’aprés les résultats du test, la p-value est inférieure à tous les seuils conventionnels .Notre série des rendements logarithmiques négatifs est donc stationnaire. C’est cette série qui sera utilisé pour estimer notre modèle. Nous allons par ailleurs vérifier s’il n’existe pas une autocorrélation de notre série en traçant l’autocorrélogramme partiel(ACF).

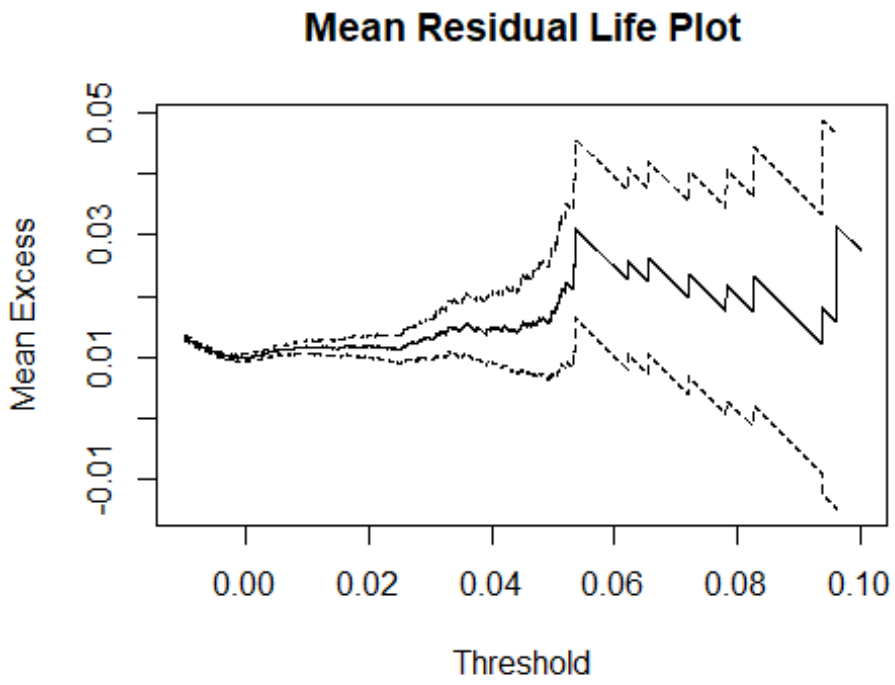
```
acf(na.omit(rend_negatif$QQQ.Close))
```



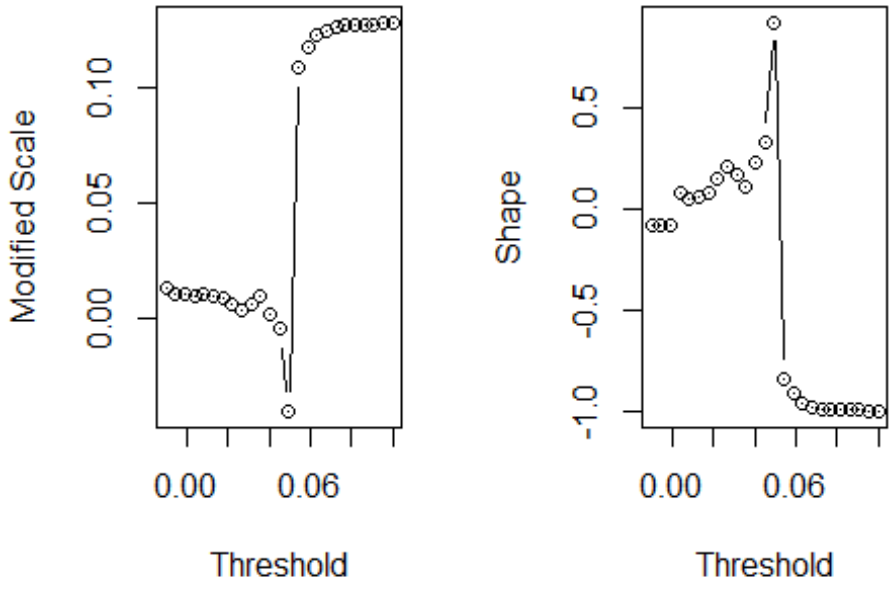
La majorité des pics n'étant pas significatifs, alors nous ne pouvons pas soupçonner la présence d'une autocorrélation.

Identifions les valeurs de seuils sensibles afin que les dépassements puisse raisonnablement être considéré comme GPD. Nous allons utiliser pour cela les fonctions "mrlplot" et "tcplot"

```
mrlplot(rend_negatif$QQQ.Close, c(-0.01, 0.10))
```




```
par(mfrow = c(1, 2))
tcplot(rend_negatif$QQQ.Close, c(-0.01, 0.10),std.err=F)
```



Sur la base des tracés précédents, un seuil autour de 0,04 devrait être bon.

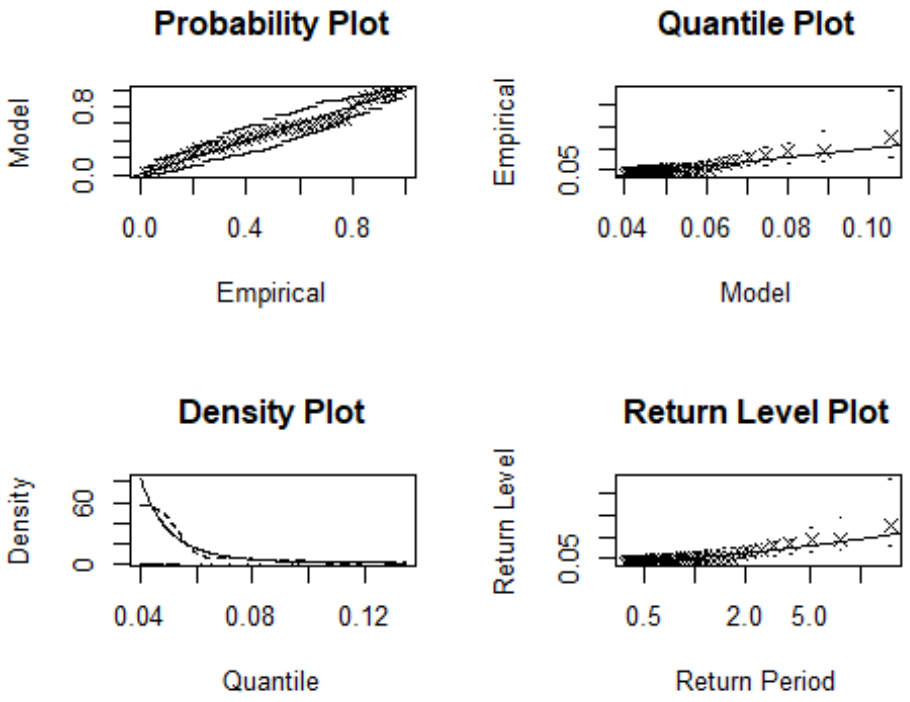
```
thresh1=0.04
```

Estimation du modèle

```
(model <- fpot(rend_negatif$QQQ.Close, thresh1, npp = 250))

##
## Call: fpot(x = rend_negatif$QQQ.Close, threshold = thresh1, npp = 250)
## Deviance: -252.6477
##
## Threshold: 0.04
## Number Above: 39
## Proportion Above: 0.0106
##
## Estimates
##   scale    shape
## 0.01165  0.21461
##
## Standard Errors
##    scale     shape
## 0.002832  0.190410
##
## Optimization Information
##   Convergence: successful
##   Function Evaluations: 39
##   Gradient Evaluations: 6
```

Graphiques du modèle estimé

```
par(mfrow = c(2,2))
plot(model)
```



D’après le Quantile plot, nous pouvons dire que le modèle semble bon.

L’interval de confiance des paramètres estimés basé sur la normalité asymptotique

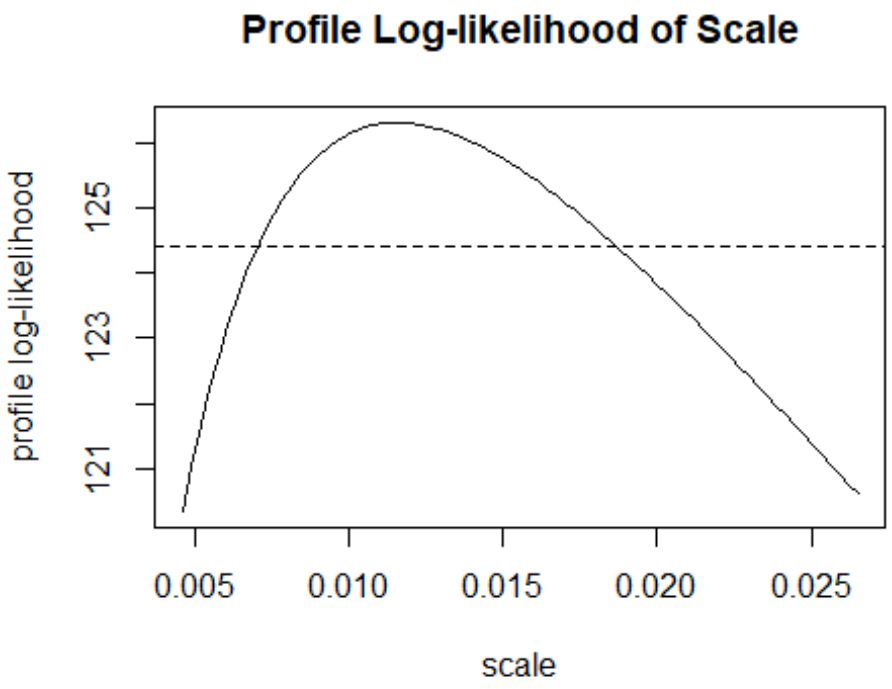
```
confint(model)
```

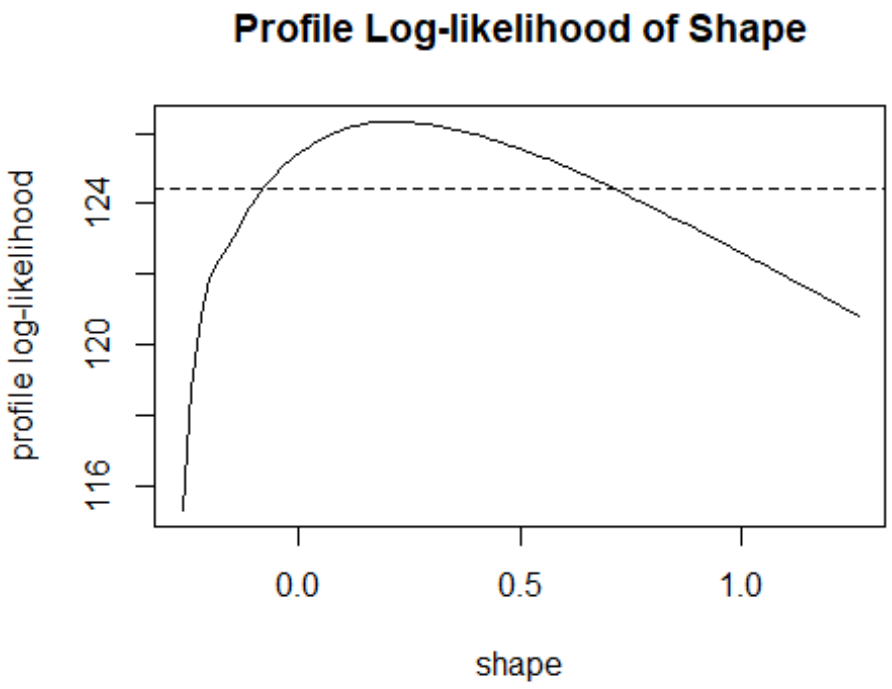
```
##              2.5 %     97.5 %
## scale  0.006104592 0.01720476
## shape -0.158588673 0.58780676
```

L’interval de confiance basé sur la vraisemblance profilée

```
plot(profile(model))
```

```
## [1] "profiling scale"
## [1] "profiling shape"
```





L'interval de confiance basé sur la vraisemblance profilée sera préféré à celui basé sur la normalité asymptotique à cause de sa forte asymétrie.

Estimons un modèle exponnentiel($\zeta = 0$) pour la comparer à notre modèle précédent.

```
(model1 <- fpot(rend_negatif$QQQ.Close, thresh1, shape = 0, npp = 250))

##
## Call: fpot(x = rend_negatif$QQQ.Close, threshold = thresh1, npp = 250,      shape = 0)
## Deviance: -250.8419
##
## Threshold: 0.04
## Number Above: 39
## Proportion Above: 0.0106
##
## Estimates
##   scale
## 0.01476
##
## Standard Errors
##    scale
## 0.002331
##
## Optimization Information
##   Convergence: successful
##   Function Evaluations: 25
##   Gradient Evaluations: 1
```

Effectuons un test anova pour comparer les deux modèles. L'hypothèse nulle du test est que le second modèle (model1) est meilleur que le premier (model)

```
anova(model, model1)

## Analysis of Deviance Table
##
##        M.Df Deviance Df  Chisq Pr(>chisq)
## model     2  -252.65
## model1    1  -250.84  1 1.8058      0.179
```

Au seuil de 5%, nous rejetons l'hypothèse nulle. Le premier modèle (model) est donc préféré au second (model1)

Estimation des niveaux de retours pour 2,10 et 100 ans avec le meilleur modèle.

```
vec_ret <- c(2, 10, 100)
moy <- mean(rend_negatif$QQQ.Close > thresh1, na.rm = TRUE)
qgpd(1 - 1 / (moy * 250 * vec_ret), thresh1, model$par["scale"], model$par["shape"])

## [1] 0.06334367 0.09537864 0.16547994
```