

Value at risk : Projet n°1
Caractéristiques des rendements logarithmiques de l'action NOC

BAJI Mohamed Anas

16 octobre 2024

Contents

I-Présentation des données	3
II- Etudes des 8 caractéristiques des rendements logarithmiques	5
1) Propriété 1 : Asymétrie perte/gain	5
2) Propriété 2 : Queues de distribution épaisses	5
3) Propriété 3 : Autocorrélations des carrés des rendements fortes et faibles pour les rendements	6
4) Propriété 4 : Clusters de volatilité	13
5) Propriété 5 : Queues épaisse conditionnelles	14
6) Propriété 6 : Effet de levier	16
7) Propriété 7 : La saisonnalité	18
8) Propriété 8 : Stationnarité	20
1.1) Dickey-Fuller (DF)	21
1.2) Dickey-Fuller Augmenté (ADF)	25
1.3) Zivot-Andrews (ZA)	28
1.4) Lee et Strazicich (LS)	32
III-Conclusion	32
IV-Annexes	33
1) Propriété 1: Asymétrie Perte/Gain	34
2) Propriété 2 : Queues de distribution épaisses	34
3) Propriété 3 : Autocorrélations des carrés des rendements fortes et faibles pour les rendements	34
4) Propriété 4 : Clusters de volatilité	37
5) Propriété 5 : Queues épaisse conditionnelles	38
6) Propriété 6 : Effet de levier	40
7) Propriété 7 : La saisonnalité	40
8) Propriété 8 : Stationnarité	42
9) Conclusion	48
10) Tableau récapitulatif	48

I-Présentation des données

Nous allons étudier les huit principales propriétés des rendements logarithmiques de l'action Elbit Systems (symbole boursier : ESLT). Nous considérons le cours de l'action observé en clôture sur la période du 01/01/2015 au 16/09/2024.

Elbit Systems est une entreprise israélienne de l'armement, fondée en 1967 et basée à Haïfa. Elbit Systems se classait en 2023 au 21e rang mondial pour la production d'armement. Elle est le principal fournisseur de véhicules terrestres et de drones de l'armée israélienne. Elle est côtée au Nasdaq.

Lors de ce projet, nous allons découper l'ensemble des données en deux parties. La première partie de notre découpage sera dédiée à la phase d'estimation, et elle consistera en un ensemble de données que nous appellerons *rte*. Cette partie comprendra approximativement les deux tiers des données les plus anciennes, commençant le 01/01/2015 et se terminant le 31/12/2020. La seconde partie sera destinée au test, et elle sera nommée *rtt*. Elle contiendra le tiers restant des données.

```
op<-par(mfrow=c(3,1))
plot(datesp, pt, type='l',ylab="cours de l'action ESLT",col=3)
plot(dates,dpt,type='l',col=2,ylab="variations de l'action ESLT")
plot(dates,rt,type='l',col=1,ylab="rendement de l'action ESLT")
```

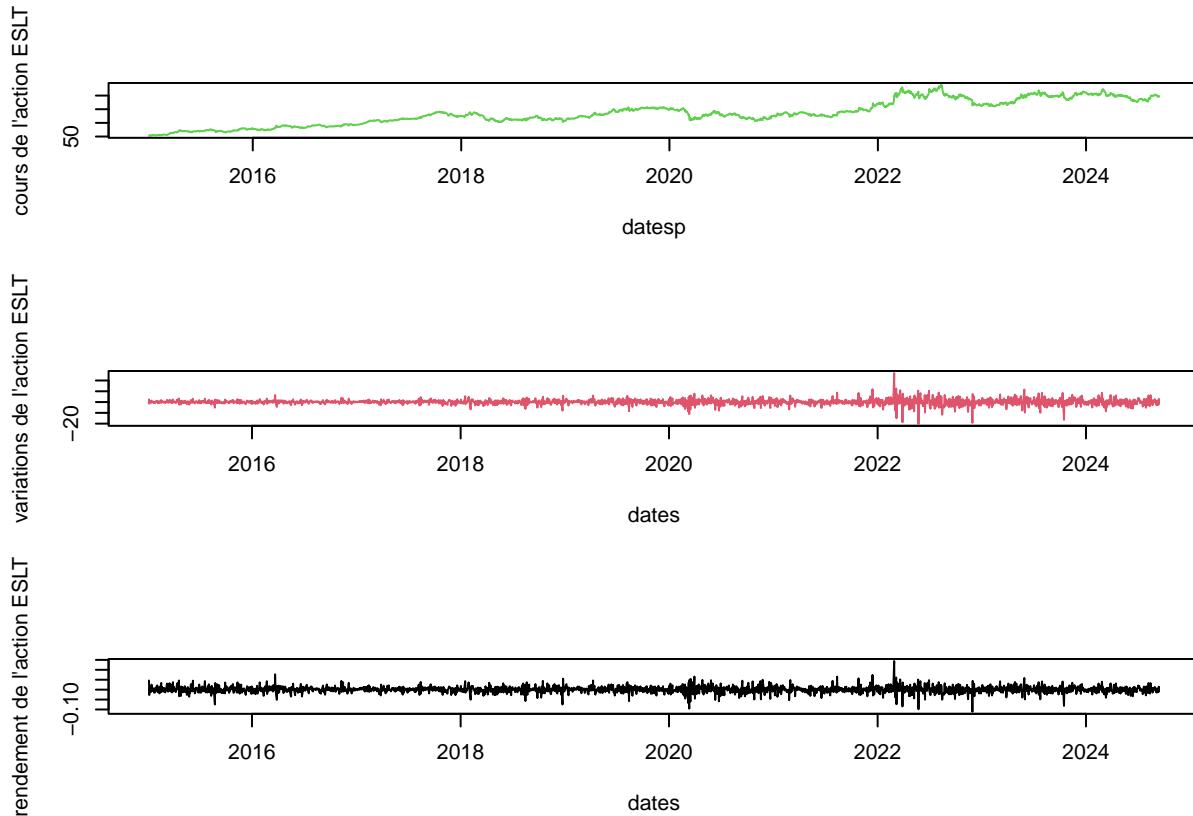


Figure 1: Valeurs, variations et rendements logarithmique de l'action NOC

```
par(op)
```

Nous remarquons que la série des cours de l'action ESLT a une tendance croissante, que ses rendements logarithmiques fluctuent autour de 0 ainsi que son rendement. A partir de 2022, les fluctuations autour de 0 des variations du cours de l'action augmentent, cela indique la volatilité croissante de l'action et nous donne un premier indice sur la présence de clusters de volatilité.

```
plot(rte, type='l', xlab="Années", ylab= "rte", main="Chronogramme de ", col="red")
```

Chronogramme de

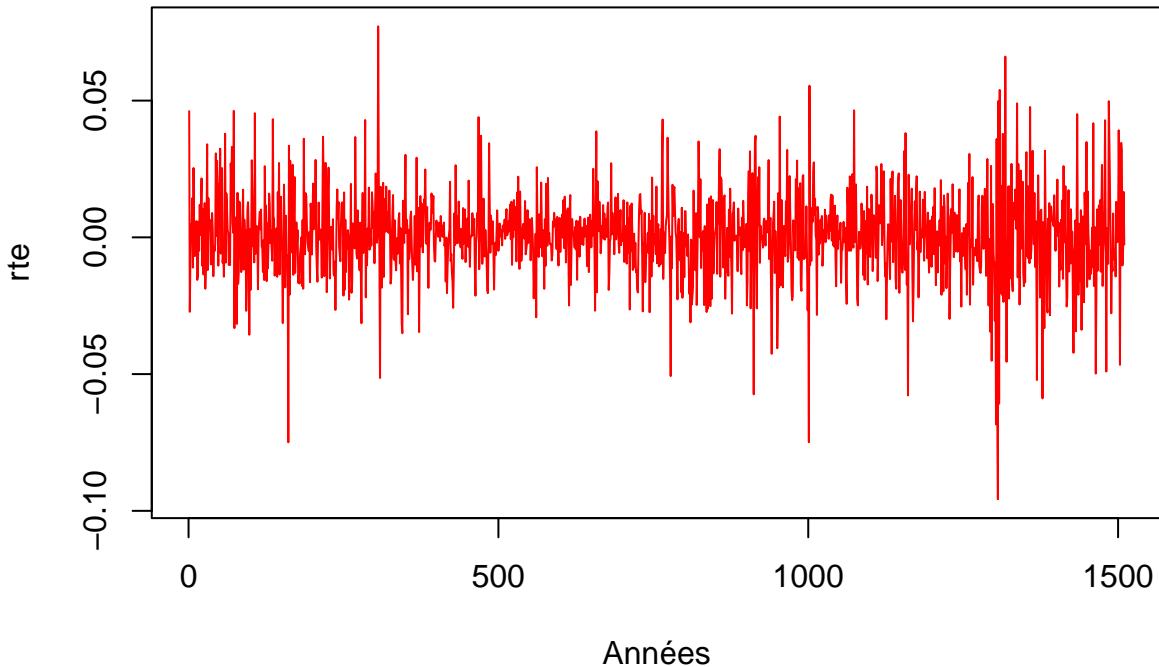


Figure 2: Chronogramme de *rte*

Calculons la moyenne de *rte* :

```
rbar<-mean(rte) #moyenne empirique  
rbar  
  
## [1] 0.0005747381  
s=sd(rte)#écart type estimé
```

Nous regardons si notre moyenne empirique, proche de 0, est statistiquement nulle, c'est à dire si l'espérance du PGD de *rte* (μ) est nulle. Pour cela nous testons : $H_0 : E(rte) = \mu = 0$ vs $H_a : E(rte) = \mu \neq 0$

- On décide en effectuant le test suivant:

```
t.test(rte)  
  
##  
##  One Sample t-test  
##  
## data: rte  
## t = 1.4346, df = 1509, p-value = 0.1516  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.0002111081 0.0013605844  
## sample estimates:  
## mean of x  
## 0.0005747381
```

La p -value vaut $0.1516 < 0.05$ donc on accepte H_0 . Donc la moyenne empirique est statistiquement nulle

II- Etudes des 8 caractéristiques des rendements logarithmiques

1) Propriété 1 : Asymétrie perte/gain

Pour tester l'hypothèse de symétrie, nous testons la nullité de la skewness qui est égale au moment centré d'ordre 3 normalisé de la distribution. Soit μ_3 , le moment centré d'ordre 3 d'une variable aléatoire : $\mu_3 = E[(X - \mu)^3]$. On veut tester si la skewness est nulle :

$$H_0 : E\left[\left(\frac{X - E(X)}{\sigma_X}\right)^3\right] = 0$$

vs

$$H_a : E\left[\left(\frac{X - E(X)}{\sigma_X}\right)^3\right] \neq 0$$

La skewness est significative si la p -value est < 0.05 . Effectuons le test :

```
agostino.test(rte)
```

```
##  
## D'Agostino skewness test  
##  
## data: rte  
## skew = -0.23895, z = -3.75761, p-value = 0.0001715  
## alternative hypothesis: data have a skewness
```

On trouve une p -value associée à la skewness de $1.715\text{e-}04 < 0.05$, on rejette donc H_0 et la skewness est significative. De plus $\text{skew} = -0.23895 < 0$, donc la probabilité de gains est supérieure à la probabilité de pertes. Ainsi, les gains sont faibles mais les pertes plus rares sont élevées.

2) Propriété 2 : Queues de distribution épaisse

Les queues de la distribution des rendements logarithmiques sont plus épaisse que celles d'une distribution normale. Dans de tels cas, on parle de distribution leptokurtique. Pour décrire ce phénomène, on utilise le moment centré d'ordre 4 d'une variable aléatoire, noté μ_4 , défini comme : $\mu_4 = E[(X - \mu)^4]$. On veut tester si la kurtosis vaut 3 :

$$H_0 : E\left[\left(\frac{X - E(X)}{\sigma_X}\right)^4\right] = 3$$

vs

$$H_a : E\left[\left(\frac{X - E(X)}{\sigma_X}\right)^4\right] \neq 3$$

Sous H_0 , on a la p -value du test d'Anscombe > 0.05 . Effectuons le test :

```
anscombe.test(rte)
```

```
##  
## Anscombe-Glynn kurtosis test  
##  
## data: rte  
## kurt = 6.4564, z = 11.2781, p-value < 2.2e-16  
## alternative hypothesis: kurtosis is not equal to 3
```

On trouve une $p-value < 2.2e-16$, ce qui est inférieur à 0.05 donc le kurtosis est significatif et égal à 6.4564. La distribution est donc leptokurtique. Ainsi les queues de distribution sont plus épaisse que celles d'une loi normale.

3) Propriété 3 : Autocorrélations des carrés des rendements fortes et faibles pour les rendements

- Les coefficients d'auto-corrélation totale et le corréogramme

La mesure du degré d'autocorrélation dans une série se fait à l'aide des coefficients d'autocorrélations totales, notés $\rho(1), \rho(2), \dots, \rho(k)$ définis par :

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)}$$

L'autocorrélation totale au retard k est définie par l'expression suivante : $\gamma(k) = cov(rte_t, rte_{t-k}) = E[(rte_t - \mu)(rte_{t-k} - \mu)]$

La suite des ρ forme la fonction d'autocorrélation totale et le graphique associé se nomme corréogramme. Si la séquence $\{rte\}$ est une série indépendante et identiquement distribuée qui satisfait $E(rte_t^2) < \infty$, alors $\hat{\rho}(h)$ suit asymptotiquement une distribution normale :

$$\sqrt{T}\hat{\rho}(h) \sim \mathcal{N}(0, 1)$$

Cela signifie que l'intervalle $[-1.96/\sqrt{T}, 1.96/\sqrt{T}]$ forme un intervalle de confiance asymptotique à 95% pour les $\rho(h)$ autour de 0. Les corréogrammes permettent de vérifier s'il existe des autocorrélations faibles dans les rendements et des autocorrélations fortes dans les rendements au carré.

```
op<-par(mfrow=c(2,2))
Acf(rte,main='ACF du rendement logarithmique')
Pacf(rte,main="PACF du rendement logarithmique")
Acf(rte^2,main='ACF du rendement logarithmique au carre')
Pacf(rte^2,main="PACF du rendement logarithmique au carré")

par(op)
```

D'après le graphique nous remarquons qu'il y a de faibles autocorrélations aux ordres 6,13 et 31 et une forte autocorrélation à l'ordre 9 dans les rendements et de fortes autocorrélations aux ordres 1, 2, 3, 4, 5, 9, 10, 11, 12... dans les rendements au carré. On conclut alors qu'il y a présence d'autocorrélations faibles dans les rendements avec une autocorrélation forte à l'ordre 9 et présence d'autocorrélations fortes dans les rendements au carré.

- Une alternative chiffrée : La statistique de Ljung-Box

On teste : $H_0 : \rho(k)k = 0$ à K vs H_a : $(k) \neq 0$ pour au moins une valeur de k . La statistique de Ljung-Box s'écrit :

$$Q_K = T(T+2) \sum_{k=1}^K \frac{\hat{\rho}(k)^2}{T-k}$$

et sous H_0 on a :

$$Q_K \xrightarrow{T \rightarrow +\infty} \chi^2(K)$$

La règle de décision est si la $p-value$ est inférieure à 0.05 on rejette H_0 .

```
pvaluesrt =rep(0,30)
pvaluesrt2 =rep(0,20)
for (i in 1:30 ) {
  pvaluesrt[i] = Box.test(rte,lag=i,type="Ljung-Box")$p.value
  pvaluesrt2[i] = Box.test(rte^2,lag=i,type="Ljung-Box")$p.value
}
```

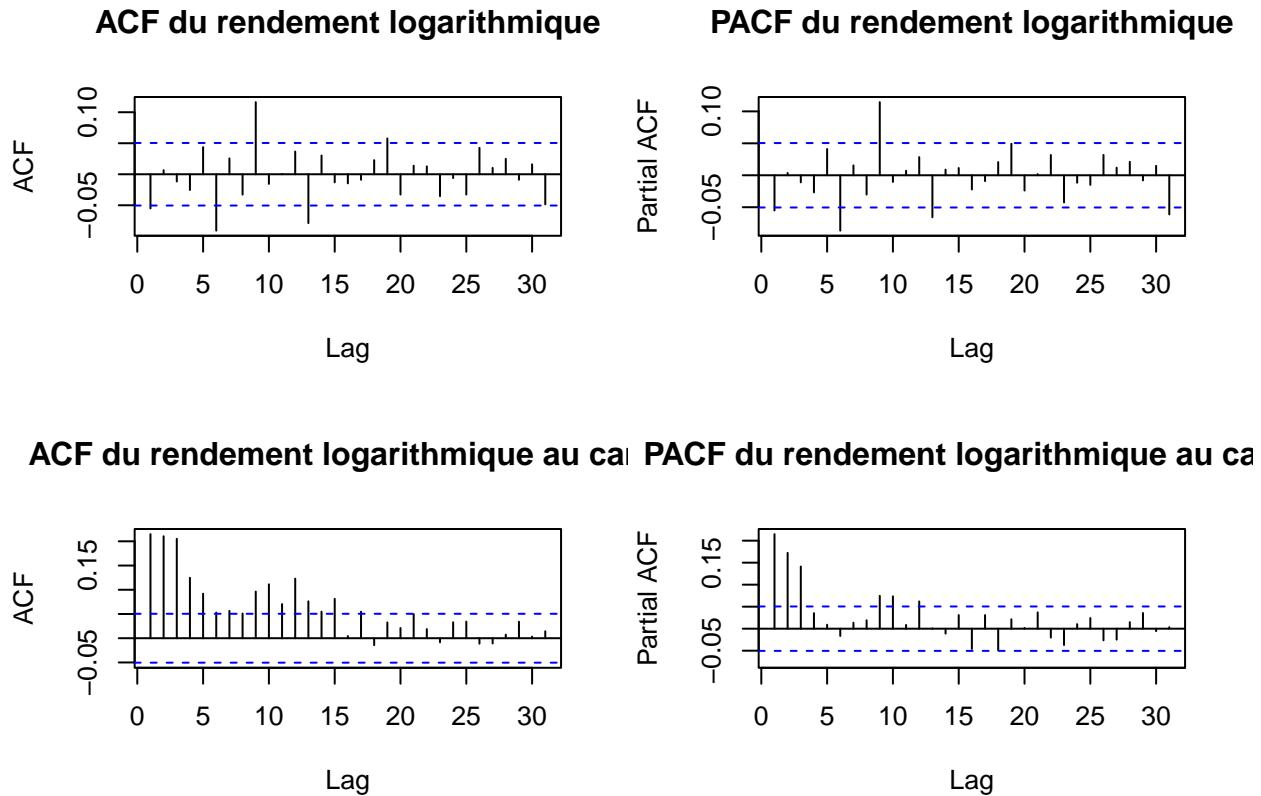


Figure 3: Corrélogrammes de rte et rte^2

Affichons les $p - value$ associées aux rendements au carré du test de Ljung-Box :

```
pvaluesrt2
```

```
## [1] 1.110223e-16 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [6] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [11] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [16] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [21] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [26] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
```

On remarque qu'elles sont quasiment toutes nulles, on conclut alors la présence d'autocorrélations dans les rendements au carré.

Affichons les $p - value$ associées aux rendements du test de Ljung-Box :

```
pvaluesrt
```

```
## [1] 3.200278e-02 9.709381e-02 1.816177e-01 2.107597e-01 1.194489e-01
## [6] 1.616637e-03 2.257158e-03 2.345469e-03 1.139677e-06 2.293665e-06
## [11] 5.110005e-06 4.815147e-06 2.307999e-07 2.869778e-07 5.471356e-07
## [16] 9.845102e-07 1.862925e-06 2.675496e-06 7.666468e-07 7.972413e-07
## [21] 1.345606e-06 2.260528e-06 2.062125e-06 3.595995e-06 3.558417e-06
## [26] 2.395321e-06 3.913751e-06 4.834314e-06 7.776696e-06 1.128067e-05
```

Plusieurs $p - value$ sont supérieures à 5%. On rejette H_0 et on conclut à la présence d'autocorrélation dans rte . Afin de modéliser cette caractéristique nous utiliserons un modèle ARMA(p,q).

```

eacf(rte)

## AR/MA
##  0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o x o o x o o o x o
## 1 x o o o o x o o x o o o x o
## 2 x x o o o x o o x o o o x o
## 3 x x x o o o o o x o o o o o
## 4 x x x x o o o o x o o o x o
## 5 x x x x x o o o o x o o o o o
## 6 x x x o o x o o x o o o o o
## 7 x o x x o x o o x o o o o o

```

A partir de l'eacf, On sélectionne les valeurs qui correspondent au sommet dans le coin supérieur gauche d'un triangle qui est rectangle en haut à droite et ouvert à droite ; triangle qui ne contient que des ronds et aucune croix.

On trouve p=5 et q=9.

On estime le modèle ARMA(5,9)

```

reg1=Arima(rte, order=c(5,0,9))
coeftest(reg1)

```

```

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1      -0.40724949  0.22247116 -1.8306 0.0671644 .
## ar2       0.14761617  0.25984274  0.5681 0.5699683
## ar3       0.08776834  0.21773563  0.4031 0.6868777
## ar4      -0.29250393  0.22822307 -1.2817 0.1999626
## ar5       0.09910869  0.18303133  0.5415 0.5881734
## ma1       0.36540074  0.22073866  1.6554 0.0978526 .
## ma2      -0.16413657  0.25233933 -0.6505 0.5153953
## ma3      -0.09176187  0.21318723 -0.4304 0.6668839
## ma4       0.27425040  0.22232535  1.2336 0.2173690
## ma5      -0.07897825  0.17596550 -0.4488 0.6535557
## ma6      -0.06191654  0.02786469 -2.2220 0.0262804 *
## ma7      -0.02310134  0.03269325 -0.7066 0.4798096
## ma8      -0.01301261  0.03008436 -0.4325 0.6653510
## ma9       0.12304142  0.03247118  3.7892 0.0001511 ***
## intercept 0.00057616  0.00038469  1.4978 0.1341967
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Les coefficients ne sont pas tous significatifs. On va fixer à 0 la valeur des coefficients qui ne sont pas significatifs un par un en essayant de conserver un côté AR et un côté MA.

```

reg2=Arima(rte, order=c(5,0,9), fixed=c(NA,NA,0,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA))
coeftest(reg2)

```

```

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1      -0.41849933  0.25776927 -1.6235 0.1044735

```

```

## ar2      0.09331149  0.26123366  0.3572 0.7209455
## ar4     -0.32556267  0.16533083 -1.9692 0.0489348 *
## ar5      0.10998213  0.18195598  0.6044 0.5455487
## ma1      0.37686967  0.25628407  1.4705 0.1414222
## ma2     -0.11094265  0.25275350 -0.4389 0.6607078
## ma3     -0.00716145  0.02996257 -0.2390 0.8110954
## ma4      0.30324004  0.16444111  1.8441 0.0651737 .
## ma5     -0.09205395  0.17532186 -0.5251 0.5995437
## ma6     -0.06242117  0.02768604 -2.2546 0.0241579 *
## ma7     -0.02281366  0.03542484 -0.6440 0.5195743
## ma8     -0.01604494  0.03157488 -0.5082 0.6113444
## ma9      0.11744954  0.03041752  3.8612 0.0001128 ***
## intercept 0.00057917  0.00038100  1.5201 0.1284745
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
reg3=Arima(rte, order=c(5,0,9), fixed=c(NA,NA,0,NA,NA,NA,NA,0,NA,NA,NA,NA,NA,NA))
coeftest(reg3)

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1     -0.39401058  0.24350041 -1.6181 0.1056388
## ar2      0.13204225  0.24451239  0.5400 0.5891814
## ar4     -0.34189803  0.17005998 -2.0105 0.0443830 *
## ar5      0.11186173  0.18672508  0.5991 0.5491250
## ma1      0.35185096  0.24196973  1.4541 0.1459154
## ma2     -0.14713225  0.24087368 -0.6108 0.5413138
## ma4      0.32114235  0.16785559  1.9132 0.0557217 .
## ma5     -0.09531899  0.17961351 -0.5307 0.5956340
## ma6     -0.06174377  0.02737963 -2.2551 0.0241271 *
## ma7     -0.01879373  0.03239606 -0.5801 0.5618310
## ma8     -0.01436085  0.03115553 -0.4609 0.6448412
## ma9      0.11620525  0.03007981  3.8632 0.0001119 ***
## intercept 0.00057793  0.00038436  1.5036 0.1326784
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
reg4=Arima(rte, order=c(5,0,9), fixed=c(NA,NA,0,NA,NA,NA,NA,0,NA,NA,NA,NA,0,NA,NA))
coeftest(reg4)

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1     -0.37453563  0.21811292 -1.7172 0.08595 .
## ar2      0.17150918  0.21074380  0.8138 0.41574
## ar4     -0.34529801  0.16636300 -2.0756 0.03793 *
## ar5      0.12822797  0.18351695  0.6987 0.48472
## ma1      0.33238497  0.21611529  1.5380 0.12405
## ma2     -0.18637192  0.20739614 -0.8986 0.36885
## ma4      0.32709792  0.16545466  1.9770 0.04805 *
## ma5     -0.10993360  0.17751436 -0.6193 0.53572
## ma6     -0.06335575  0.02731237 -2.3197 0.02036 *

```

```

## ma7      -0.01428730  0.02907400 -0.4914   0.62314
## ma9       0.11840154  0.02998943  3.9481 7.877e-05 ***
## intercept 0.00057750  0.00039046  1.4790   0.13913
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
reg5=Arima(rte, order=c(5,0,9), fixed=c(NA,NA,0,NA,NA,NA,NA,0,NA,NA,NA,0,0,NA,NA))
coeftest(reg5)

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1      -0.33165746  0.19840871 -1.6716  0.09461 .
## ar2       0.19080700  0.20439042  0.9335  0.35054
## ar4      -0.33146541  0.17506864 -1.8933  0.05831 .
## ar5       0.14658295  0.18973019  0.7726  0.43977
## ma1       0.29029698  0.19629102  1.4789  0.13916
## ma2      -0.20682693  0.19981488 -1.0351  0.30063
## ma4       0.31377178  0.17392680  1.8040  0.07122 .
## ma5      -0.13029839  0.18316345 -0.7114  0.47685
## ma6      -0.06067522  0.02677940 -2.2657  0.02347 *
## ma9       0.11663637  0.02966996  3.9311 8.455e-05 ***
## intercept 0.00057528  0.00039410  1.4597  0.14436
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
reg6=Arima(rte, order=c(5,0,9), fixed=c(NA,NA,0,NA,NA,NA,NA,0,NA,0,NA,0,0,NA,NA))
coeftest(reg6)

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1      -0.37377958  0.24451130 -1.5287 0.126344
## ar2       0.20898308  0.33925991  0.6160 0.537897
## ar4      -0.44777069  0.16949427 -2.6418 0.008247 **
## ar5       0.01295323  0.03004350  0.4311 0.666360
## ma1       0.33331427  0.24024238  1.3874 0.165317
## ma2      -0.22504873  0.33293362 -0.6760 0.499068
## ma4       0.43051228  0.16623969  2.5897 0.009606 **
## ma6      -0.06036670  0.02726635 -2.2140 0.026831 *
## ma9       0.11349381  0.03509197  3.2342 0.001220 **
## intercept 0.00057599  0.00039313  1.4651 0.142882
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
reg7=Arima(rte, order=c(5,0,9), fixed=c(NA,0,0,NA,NA,NA,NA,0,NA,0,NA,0,0,NA,NA))
coeftest(reg7)

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1      -0.51316926  0.13012350 -3.9437 8.023e-05 ***
## ar4      -0.35983594  0.08451308 -4.2578 2.065e-05 ***

```

```

## ar5      0.01681907  0.03123709  0.5384  0.590278
## ma1      0.47067775  0.12948557  3.6350  0.000278 ***
## ma2     -0.02064078  0.02724102 -0.7577  0.448625
## ma4      0.34604689  0.08204004  4.2180  2.465e-05 ***
## ma6     -0.05374162  0.02634435 -2.0400  0.041354 *
## ma9      0.12144105  0.02978097  4.0778  4.546e-05 ***
## intercept 0.00057652  0.00039672  1.4532  0.146158
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
reg8=Arima(rte, order=c(5,0,9), fixed=c(NA,0,0,0,NA,NA,NA,0,NA,0,NA,0,0,NA,NA))
coeftest(reg8)

```

```

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1     -0.21182733 0.15925450 -1.3301 0.183479
## ar5      0.03890490 0.02615762  1.4873 0.136929
## ma1      0.16626287 0.15813644  1.0514 0.293080
## ma2     -0.00639359 0.02566917 -0.2491 0.803302
## ma4     -0.01219228 0.02647882 -0.4605 0.645190
## ma6     -0.07199857 0.02703928 -2.6627 0.007751 **
## ma9      0.11102688 0.02559448  4.3379 1.438e-05 ***
## intercept 0.00056855 0.00040069  1.4189 0.155914
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
reg9=Arima(rte, order=c(5,0,9), fixed=c(NA,0,0,0,NA,NA,0,0,NA,0,NA,0,0,NA,NA))
coeftest(reg9)

```

```

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1     -0.20944369 0.16000367 -1.3090 0.190537
## ar5      0.03892734 0.02615100  1.4886 0.136603
## ma1      0.16526147 0.16026515  1.0312 0.302459
## ma4     -0.01240606 0.02643104 -0.4694 0.638802
## ma6     -0.07215511 0.02705837 -2.6666 0.007661 **
## ma9      0.11147463 0.02547313  4.3762 1.208e-05 ***
## intercept 0.00056831 0.00040335  1.4090 0.158850
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
reg10=Arima(rte, order=c(5,0,9), fixed=c(NA,0,0,0,NA,NA,0,0,0,0,NA,0,0,NA,NA))
coeftest(reg10)

```

```

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1     -0.22304620 0.15384636 -1.4498 0.147115
## ar5      0.04100743 0.02590229  1.5832 0.113385
## ma1      0.17938972 0.15417953  1.1635 0.244622
## ma6     -0.07082290 0.02677427 -2.6452 0.008165 **

```

```

## ma9      0.11354284  0.02508728  4.5259 6.014e-06 ***
## intercept 0.00057706  0.00040944  1.4094  0.158724
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
reg11=Arima(rte, order=c(5,0,9), fixed=c(NA,0,0,0,NA,0,0,0,0,NA,0,0,NA,NA))
coeftest(reg11)

## 
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1      -0.04561973  0.02580793 -1.7677 0.077117 .
## ar5       0.03968844  0.02581228  1.5376 0.124151
## ma6      -0.08103976  0.02476449 -3.2724 0.001066 **
## ma9       0.11160929  0.02511383  4.4441 8.825e-06 ***
## intercept 0.00057265  0.00040587  1.4109  0.158267
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
reg12=Arima(rte, order=c(5,0,9), fixed=c(NA,0,0,0,NA,0,0,0,0,NA,0,0,NA,0))
coeftest(reg12)

## 
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1     -0.044326  0.025810 -1.7174 0.085903 .
## ar5      0.041013  0.025812  1.5889 0.112073
## ma6     -0.079970  0.024743 -3.2320 0.001229 **
## ma9      0.112751  0.025081  4.4954 6.944e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
reg13=Arima(rte, order=c(5,0,9), fixed=c(NA,0,0,0,0,0,0,0,0,NA,0,0,NA,0))
coeftest(reg13)

## 
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1     -0.045156  0.025826 -1.7485 0.080379 .
## ma6     -0.081291  0.024753 -3.2841 0.001023 **
## ma9      0.113005  0.025182  4.4875 7.205e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
reg14=Arima(rte, order=c(5,0,9), fixed=c(0,0,0,0,0,0,0,0,0,NA,0,0,NA,0))
coeftest(reg14)

## 
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ma6     -0.083277  0.024643 -3.3793 0.0007268 ***
## ma9      0.114745  0.025167  4.5594 5.13e-06 ***
## ---

```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ce modèle sert donc à modéliser l'autocorrélation détectée dans rte. On vérifie que les aléas sont des bruits blancs. On commence par vérifier que les aléas ont une espérance statistiquement nulle. $H_0 : E(\epsilon) = 0$ vs $H_a : E(\epsilon) \neq 0$

```
residu=reg14$res
t.test(residu)
```

```
##
##  One Sample t-test
##
## data:  residu
## t = 1.4063, df = 1509, p-value = 0.1598
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.0002200859  0.0013350320
## sample estimates:
##   mean of x
## 0.000557473
```

On trouve une p-value > 0.05. Donc on accepte H_0 donc l'espérance des aléas est bel est bien nulle. On passe au test de l'autocorrélation dans le résidu. Testons maintenant : H_0 : Absence d'autocorrélation jusqu'à l'ordre 20 vs H_a Présence d'autocorrélation

```
residuv=(residu-mean(residu))/sd(residu)
K<-20
tmp<-rep(0,K)
for(i in 1:K){
  tmp[i]<-Box.test(residuv,lag=i,type="Ljung-Box")$p.value
}
tmp
## [1] 0.07437319 0.20352379 0.35407773 0.39619824 0.25930311 0.36390671
## [7] 0.43499170 0.45618346 0.55687565 0.59259831 0.67682238 0.56142139
## [13] 0.12549140 0.13272094 0.16789100 0.20622806 0.24617055 0.25559699
## [19] 0.15628802 0.15452226
```

Toutes les p-value sont supérieures à 0.05. Notre modèle ARMA(5,9) n'est pas auto-corrélu.

4) Propriété 4 : Clusters de volatilité

Nous observons empiriquement que de fortes variations dans les rendements sont généralement suivies de fortes variations ultérieures. Cela se traduit par un phénomène de regroupement des extrêmes en clusters ou en paquets de volatilité. Ce phénomène remet en question l'hypothèse d'homoscédasticité conditionnelle, qui suppose que la variance de l'erreur au temps t ne dépend pas des variations au carré des erreurs des périodes précédentes.

Afin de vérifier cette hypothèse, nous utilisons le test ARCH d'Engle (1982). Si nous notons ϵ_t comme étant le t -ième résidu d'un modèle ARMA(p,q) représentant l'équation de la moyenne conditionnelle de r_t , et e comme étant les résidus associés à son estimation, alors le modèle ARCH(m) représentant l'équation de la volatilité de r_t est formulé comme suit :

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2 + \dots + \alpha_m \epsilon_{t-m}^2$$

On teste $H_0 : \alpha_1 = \alpha_2 = \dots = \alpha_m = 0$ (homoscédasticité conditionnelle) vs $H_a : \alpha_i \neq 0$ avec $i \neq 0$.

Pour effectuer ce test nous estimons par MCO :

$$e_t^2 = a + c_1 e_{t-1}^2 + \dots + c_p e_{t-p}^2 + erreurs$$

puis nous multiplions le coefficient de détermination (le R^2) de cette dernière régression avec N afin d'obtenir $LM = NxR^2$. Sous H_0 , $LM \sim \chi^2(m)$.

```
LM1<-ArchTest(as.numeric(rte),lag=1)
LM1
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  as.numeric(rte)
## Chi-squared = 70.119, df = 1, p-value < 2.2e-16
```

La $p - value = 2.2e-16 < 0.05$, on rejette donc H_0 et donc il y a présence de clusters de volatilité à l'ordre 1 dans *rte*.

```
LM2<-ArchTest(as.numeric(rte),lag=2)
LM2
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  as.numeric(rte)
## Chi-squared = 112.45, df = 2, p-value < 2.2e-16
```

La $p - value < 0.05$, on rejette donc H_0 et donc il y a présence de clusters de volatilité à l'ordre 2 dans *rte*.

```
LM20<-ArchTest(as.numeric(rte),lag=20)
LM20
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  as.numeric(rte)
## Chi-squared = 174.11, df = 20, p-value < 2.2e-16
```

La $p - value < 0.05$, on rejette donc H_0 et donc il y a présence de clusters de volatilité à l'ordre 20 dans *rte*.

```
LM40<-ArchTest(as.numeric(rte),lag=40)
LM40
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  as.numeric(rte)
## Chi-squared = 183.28, df = 40, p-value < 2.2e-16
```

La $p - value < 0.05$, on rejette donc H_0 . Il y a présence de clusters de volatilité dans *rte*.

5) Propriété 5 : Queues épaissees conditionnelles

Nous souhaitons modéliser les clusters de volatilité de notre série, pour cela nous estimons un modèle GARCH(1,1) sur notre série *rte* normalisée.

```
rte_centre=(rte-mean(rte))/sd(rte)
volat=garch(rte_centre,order=c(1,1))
```

```

##
## ***** ESTIMATION WITH ANALYTICAL GRADIENT *****
##
##
##      I      INITIAL X(I)          D(I)
##
##      1      9.000000e-01      1.000e+00
##      2      5.000000e-02      1.000e+00
##      3      5.000000e-02      1.000e+00
##
##      IT      NF      F          RELDF      PRELDF      RELDX      STPPAR      D*STEP      NPRELDF
##      0      1      7.268e+02      7.48e-03      2.73e-02      5.5e-02      2.0e+03      1.0e-01      2.71e+01
##      1      3      7.214e+02      7.48e-03      2.73e-02      5.5e-02      2.0e+03      1.0e-01      2.71e+01
##      2      5      7.192e+02      3.00e-03      4.23e-03      2.3e-02      3.1e+00      4.7e-02      3.80e+00
##      3      7      7.163e+02      4.09e-03      4.64e-03      9.0e-02      2.0e+00      1.8e-01      9.88e-01
##      4      8      7.041e+02      1.70e-02      2.14e-02      2.5e-01      2.0e+00      3.5e-01      8.47e-01
##      5      9      7.024e+02      2.42e-03      2.04e-02      2.8e-01      1.9e+00      3.5e-01      4.30e-01
##      6      11     6.982e+02      5.98e-03      1.56e-02      2.3e-02      1.0e+01      3.5e-02      2.23e+00
##      7      12     6.952e+02      4.29e-03      3.84e-03      2.3e-02      2.0e+00      3.5e-02      9.14e-01
##      8      13     6.908e+02      6.26e-03      6.15e-03      5.1e-02      2.0e+00      7.1e-02      1.64e+00
##      9      14     6.876e+02      4.70e-03      1.09e-02      7.3e-02      2.0e+00      1.4e-01      1.93e+00
##      10     15     6.796e+02      1.17e-02      3.19e-02      8.6e-02      1.7e+00      1.4e-01      1.58e-01
##      11     17     6.768e+02      4.09e-03      1.50e-02      7.6e-03      1.2e+01      1.9e-02      1.23e+00
##      12     18     6.751e+02      2.45e-03      2.05e-03      8.6e-03      2.0e+00      1.9e-02      1.70e-01
##      13     19     6.743e+02      1.18e-03      1.69e-03      1.1e-02      2.5e+00      1.9e-02      3.64e-01
##      14     21     6.711e+02      4.83e-03      6.77e-03      4.9e-02      2.0e+00      9.8e-02      3.57e-01
##      15     24     6.707e+02      5.66e-04      1.19e-03      9.5e-04      4.5e+01      2.6e-03      5.19e-02
##      16     25     6.706e+02      1.39e-04      1.55e-04      1.3e-03      2.1e+00      2.6e-03      1.66e-03
##      17     26     6.704e+02      2.74e-04      3.37e-04      2.3e-03      1.6e+00      5.1e-03      1.21e-03
##      18     27     6.703e+02      1.65e-04      2.03e-04      2.3e-03      7.1e-01      5.1e-03      2.38e-04
##      19     28     6.703e+02      1.79e-05      2.89e-05      2.3e-03      8.0e-01      5.1e-03      4.23e-05
##      20     29     6.703e+02      2.71e-06      4.85e-06      7.6e-04      0.0e+00      1.7e-03      4.85e-06
##      21     30     6.703e+02      2.68e-07      1.12e-07      9.1e-05      0.0e+00      2.0e-04      1.12e-07
##      22     31     6.703e+02      -4.66e-09      7.64e-11      4.5e-06      0.0e+00      9.3e-06      7.64e-11
##
## ***** RELATIVE FUNCTION CONVERGENCE *****
##
##      FUNCTION      6.702909e+02      RELDX      4.498e-06
##      FUNC. EVALS      31      GRAD. EVALS      22
##      PRELDF      7.637e-11      NPRELDF      7.637e-11
##
##      I      FINAL X(I)          D(I)          G(I)
##
##      1      4.467333e-02      1.000e+00      -2.726e-02
##      2      9.435179e-02      1.000e+00      -2.002e-02
##      3      8.636184e-01      1.000e+00      -3.534e-02

summary(volat)

##
## Call:
## garch(x = rte_centre, order = c(1, 1))
##
## Model:
## GARCH(1,1)

```

```

## 
## Residuals:
##      Min       1Q   Median      3Q     Max 
## -5.23592 -0.54199  0.00395  0.55571  5.93441
## 
## Coefficient(s):
##      Estimate Std. Error t value Pr(>|t|)    
## a0    0.04467   0.01042   4.288 1.80e-05 ***
## a1    0.09435   0.01344   7.020 2.22e-12 ***
## b1    0.86362   0.01868  46.241 < 2e-16 ***
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Diagnostic Tests:
## Jarque Bera Test
## 
## data: Residuals
## X-squared = 384.29, df = 2, p-value < 2.2e-16
## 
## Box-Ljung test
## 
## data: Squared.Residuals
## X-squared = 0.19005, df = 1, p-value = 0.6629
ArchTest(volat$res,lag=1)

## 
## ARCH LM-test; Null hypothesis: no ARCH effects
## 
## data: volat$res
## Chi-squared = 0.18959, df = 1, p-value = 0.6633

```

Nous avons une $p - value$ de $0.6633 > 0.05$ donc le modèle Garch a pris en compte les effets Arch et le modèle Garch permet de modéliser notre ARMA(5,9) tout en prenant en compte l'hétéroscédascticité. Nous regardons maintenant si les queues de distribution des aléas de notre ARMA-GARCH sont plus épaisses que celles d'une loi normale. Pour cela nous testons : $H_0 : kurtosis = 3$ vs $H_a : kurtosis \neq 3$ On rejette H_0 si la $p - value$ est inférieure à 0.05.

```

anscombe.test(volat$res)

## 
## Anscombe-Glynn kurtosis test
## 
## data: volat$res
## kurt = 5.4719, z = 9.5875, p-value < 2.2e-16
## alternative hypothesis: kurtosis is not equal to 3

```

On trouve une $p - value < 2.2e-16$ donc on rejette H_0 . Ainsi les queues de distribution des aléas de notre ARMA-GARCH sont plus épaisses que celles d'une loi normale.

6) Propriété 6 : Effet de levier

Il y a une asymétrie dans l'impact des valeurs passées du cours inférieures à 0 par rapport à celles supérieures à 0 sur la volatilité du cours ou du rendement logarithmique. En d'autres termes, lorsque le cours connaît des baisses, cela a tendance à entraîner une augmentation de la volatilité plus importante que celle provoquée

par des hausses de cours de même ampleur. Pour visualiser cette tendance, on peut créer un graphique représentant les prix et leur volatilité.

Pour estimer la volatilité de manière simplifiée à l'instant t , on utilise l'écart-type des 22 derniers jours :

$$\sigma_t^{(22)} = \sqrt{\frac{\sum_{i=t-22}^t (y_i - (\sum_{i=t-22}^t y_i / 22))^2}{22}}$$

Ici, y_i représente les valeurs observées au cours des 22 derniers jours, et $\sigma_t^{(22)}$ est l'estimateur de la volatilité au moment t .

```

sig<-rep(0,T)
for(t in 1:T)
{
  sig[t]<-sqrt(sum(rte[t-22]-(sum(rte[t-22]/22)))^2/22)
}
sigma=sig[24:T]*100
plot(log(pt[24:length(rte)]),type='l',col=2,axes=F,xlab="", ylab="",lwd=3)
axis(2,at=seq(3.5,6.5,by=0.25))#axe de gauche avec 3.5 += min logpt et 6.5 +=max logpt
par(new=T)
plot(sigma, col="grey",type='l',axes = F,xlab="", ylab="")
axis(4,at=seq(0,1.5,by=0.25))#axe de droite
legend("topleft", c("log(pt)","sigma"),col = c(2, 1),lty=c(1,1))
abline(v = 1270,col = 4,lwd = 4)
abline(v = 1350,col = 4,lwd = 4)
abline(v = 370,col = 3,lwd = 4)
abline(v = 510,col = 3,lwd = 4)

```

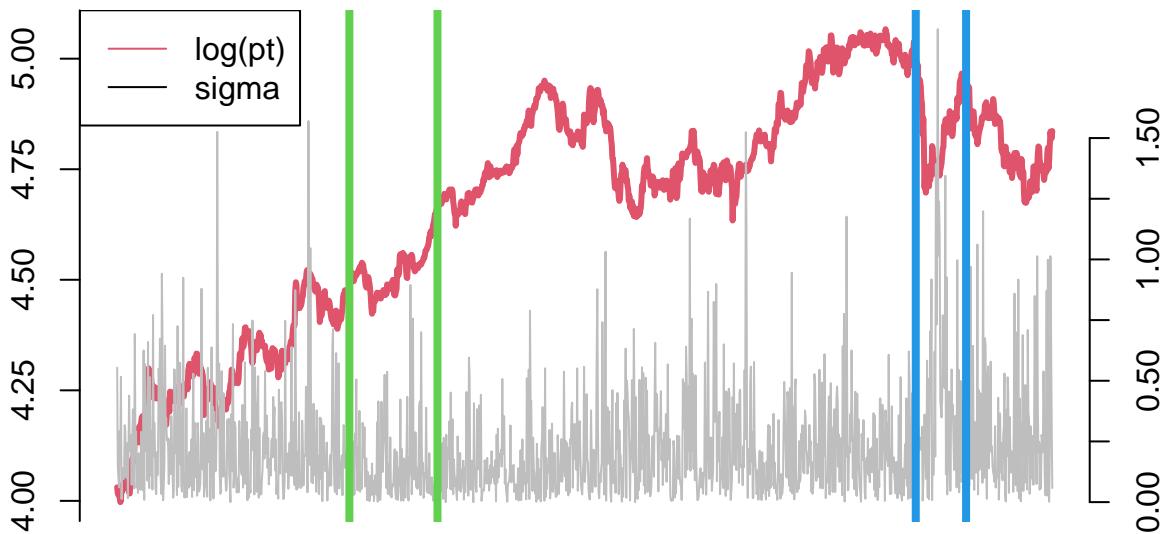


Figure 4: Logarithme de l'action ESLT journalier et écart-type récursif journalier de notre action

Graphiquement on remarque que les chutes de marché (période entre les traits bleus) ont induit une hausse de la volatilité. La volatilité y est bien plus grande que lors d'une période de croissance (période entre les traits verts) **On voit que les périodes de chute de marché sont caractérisées par une augmentation de la volatilité supérieure à celle consécutive à une hausse des cours. Notre série présente un effet de levier**

7) Propriété 7 : La saisonnalité

- Effet week-end

```

jour=format(dates[1:T], format = "%A")
tableaures <- data.frame(matrix(NA,ncol=5,nrow=4))
colnames(tableaures) <- c("lundi","mardi","mercredi","jeudi","vendredi")
rownames(tableaures) <- c("moyenne en %","ecart-type annuel en %","skewness","kurtosis")

m<-seq(from=1,to=T,by=5)
rtlun<-as.numeric(rte[m])
lundi<-mean(rtlun)

```

```

tableaures[1,1] <- lundi*100
tableaures[2,1] <- sd(rt lun)*100*sqrt(252)
tableaures[3,1] <- skewness(rt lun)
tableaures[4,1] <- kurtosis(rt lun)

m<-seq(from=2,to=T,by=5)
rtmar<-as.numeric(rte[m])
mardi<-mean(rtmar) #moyenne journaliere
tableaures[1,2] <- mardi*100 #moyenne journaliere en %
tableaures[2,2] <- sd(rtmar)*100*sqrt(252) #ecart-type annualise en %
tableaures[3,2] <- skewness(rtmar)
tableaures[4,2] <- kurtosis(rtmar)

m<-seq(from=3,to=T,by=5)
rtmer<-as.numeric(rte[m])
mer<-mean(rtmer)
tableaures[1,3] <- mer*100
tableaures[2,3] <- sd(rtmer)*100*sqrt(252)
tableaures[3,3] <- skewness(rtmer)
tableaures[4,3] <- kurtosis(rtmer)

m<-seq(from=4,to=T,by=5)
rtjeu<-as.numeric(rte[m])
jeudi<-mean(rtjeu)
tableaures[1,4] <- jeudi*100
tableaures[2,4] <- sd(rtjeu)*100*sqrt(252)
tableaures[3,4] <- skewness(rtjeu)
tableaures[4,4] <- kurtosis(rtjeu)

m<-seq(from=5,to=T,by=5)
rtven<-as.numeric(rte[m])
ven<-mean(rtven)
tableaures[1,5] <- ven*100
tableaures[2,5] <- sd(rtven)*100*sqrt(252)
tableaures[3,5] <- skewness(rtven)
tableaures[4,5] <- kurtosis(rtven)

tableaures

##                               lundi      mardi    mercredi     jeudi
## moyenne en %           0.06046845  0.04537903 -0.04710783  0.07943781
## ecart-type annuel en % 27.78501759 24.10930350 24.41669782 24.50710447
## skewness                -0.91274596  0.32281843 -0.26246477  0.12033503
## kurtosis                 5.83609239  1.26056751  3.89502863  1.50854755
##                               vendredi
## moyenne en %           0.14919161
## ecart-type annuel en % 22.51101889
## skewness                -0.05663356
## kurtosis                 2.01969111

```

On remarque que l'écart-type le plus fort est celui du lundi. Et les écarts-types diminuent progressivement jusqu'au vendredi. On peut donc attester la présence d'un effet week-end tel qu'énoncé par French et Roll (1986).

- Effet janvier

```
monthplot(rte, ylab="rendement",main="", cex.main=1,col.base=2,lwd.base=3,col='grey',lwd=0.5)
abline(h = rbar,col = 3,lwd = 1)
```

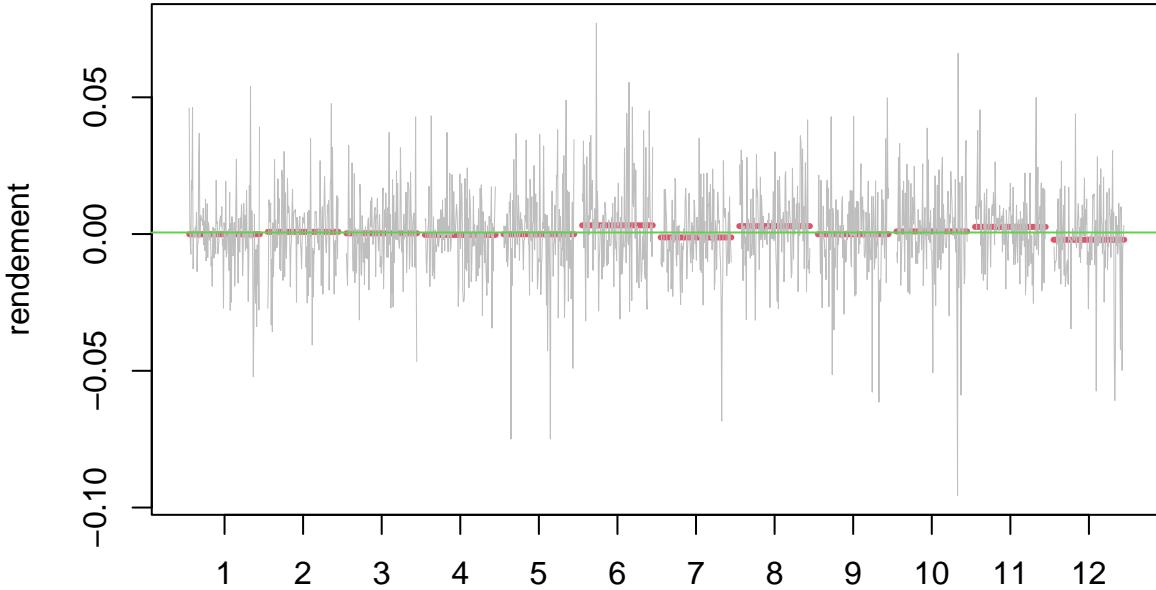


Figure 5: Rendement logarithmique par mois de l'action NOC

La moyenne des rendements semble constante sur l'année avec une légère baisse en novembre et décembre. Les mois d'Avril, Janvier et Décembre n'ont pas une moyenne plus élevée. **Il ne semble pas avoir d'effet Janvier.**

8) Propriété 8 : Stationnarité

Les processus stochastiques p_t , qui sont liés aux prix des actifs, tendent généralement à ne pas être stationnaires en ce qui concerne la stationnarité du second ordre.

Un processus stochastique $(y_t, t \in \mathbb{Z})$ est stationnaire au second ordre si :

- $\forall t \in \mathbb{Z}, E(y_t^2) = \gamma(0) < \infty$: variance finie
- $\forall t \in \mathbb{Z}, E(y_t) = \mu$: moyenne indépendante du temps
- $\forall (t, h) \in \mathbb{Z}^2, cov(y_t, y_{t+h}) = E[(y_{t+h} - \mu)(y_t - \mu)] = \gamma(h)$: fonction d'autocovariance indépendante du temps.

Les processus non stationnaires sont soit TS, soit DS.

Processus TS:

C'est un processus qui comporte une tendance déterministe : $y_t = \mu + \delta t \text{tendance}_t + u_t$ avec $\mu + \delta t \text{tendance}_t$ la partie déterministe et u_t la partie aléatoire qui est un processus ARMA stationnaire.

Un processus TS est tel que :

- $E(y_t) = \mu + \delta t \text{tendance}_t$
- $Var(y_t) = \sigma_u^2$
- Les chocs sont transitoires et s'atténuent avec le temps

- La série stationnaire y_t^* s'obtient :

$$y_t^* = y_t - \hat{\delta}_t - \hat{\mu}$$

avec $\hat{\delta}_t$ et $\hat{\mu}$ s'obtenant en estimant par MCO la régression linéaire de y_t sur une constante et une tendance déterministe.

Processus DS:

C'est un processus qui comporte une tendance stochastique. Aussi appelé marche aléatoire avec ou sans dérive (=“drift”) :

$$y_t = \delta + y_{t-1} + u_t$$

où δ est un coefficient à estimer et s'il est significatif il est appelé la dérive. Un processus DS est aussi appelé marche aléatoire lorsque u_t un bruit blanc, δ une constante appelée la dérive.

Un processus DS est tel que:

- $E(y_t|y_0) = y_0 + \delta t$
- $Var(y_t|y_0) = t\sigma_u^2$ (variance explosive)
- Les chocs sont cumulatifs et ne s'atténuent pas avec le temps
- Pour stationariser une marche aléatoire, on la différencie à l'ordre 1 :

$$\Delta y_t = y_t - y_{t-1} = \delta + u_t$$

Cette propriété fait qu'une telle série est aussi appelée stationnaire en différence (DS) ou encore intégrée d'ordre 1 et notée $I(1)$. Par analogie, une série stationnaire est $I(0)$.

Nous allons effectuer les **4 tests de racines unitaires** qui sont :

- Dickey-Fuller (DF)
- Dickey-Fuller Augmenté (ADF)
- Zivot-Andrews (ZA)
- Lee et Strazicich (LS)

Ces tests permettent de savoir si notre PGD est stationnaire, TS (Temporary stationnary) ou DS (Difference stationnary). Pour ce faire, les tests estiment par les MCO (Moindres Carrés Ordinaires) un modèle qui comprend les processus TS, DS et un processus stationnaire comme cas particulier.

1.1) Dickey-Fuller (DF)

Il existe 3 spécifications possibles pour le test de Dickey-Fuller, sachant qu'une seule est la bonne pour une série particulière.:

- “trend”
- “drift”
- “none”

Nous allons commencer par la spécification “**trend**” qui consiste à estimer par les MCO :

$$\Delta X_t = (\rho - 1)X_{t-1} + \beta_0 + \beta_1 t \text{tendance}_t + \epsilon_t$$

puis à tester :

$$H_0 : \rho - 1 = 0 \text{ et } \beta_1 = 0 \quad vs \quad H_a : |\rho| < 1 \text{ et } \beta_1 \neq 0$$

Nous testons d'abord la significativité de β_1 pour vérifier que la spécification soit la bonne :

$$H_0 : \beta_1 = 0 \quad vs \quad H_a : \beta_1 \neq 0$$

L'hypothèse nulle est rejetée au risque α de 5% si la $p-value$ associée à β_1 est inférieure à α .

```
summary(ur.df(rte,type="trend",lag=0))

##
## #####
## # Augmented Dickey-Fuller Test Unit Root Test #
## #####
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.099070 -0.007917  0.000077  0.008017  0.076104
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.577e-03 7.997e-04 1.972 0.0488 *
## z.lag.1     -1.057e+00 2.566e-02 -41.179 <2e-16 ***
## tt          -1.324e-06 9.169e-07 -1.444 0.1489
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0155 on 1506 degrees of freedom
## Multiple R-squared: 0.5296, Adjusted R-squared: 0.529
## F-statistic: 847.9 on 2 and 1506 DF, p-value: < 2.2e-16
##
##
## Value of test-statistic is: -41.1794 565.254 847.8777
##
## Critical values for test statistics:
##      1pct 5pct 10pct
## tau3 -3.96 -3.41 -3.12
## phi2  6.09  4.68  4.03
## phi3  8.27  6.25  5.34
```

On observe que la $p-value$ associée à β_1 (tt) est de $0.1489 > 0.05$. Nous ne rejetons pas H_0 avec un risque de 5%, le coefficient β_1 n'est donc pas significatif. **Nous en concluons que la spécification “trend” n'est pas adaptée à notre série, et passons au test de la spécification “drift”.**

La spécification “drift” consiste à estimer par les MCO :

$$\Delta X_t = (\rho - 1)X_{t-1} + \beta_0 + \epsilon_t$$

puis à tester :

$$H_0 : \rho - 1 = 0 \text{ et } \beta_0 = 0 \quad vs \quad H_a : |\rho| < 1 \text{ et } \beta_0 \neq 0$$

Nous commençons par tester la significativité de β_0 pour vérifier que la spécification soit la bonne :

$$H_0 : \beta_0 = 0 \quad vs \quad H_a : \beta_0 \neq 0$$

L'hypothèse nulle est rejetée au risque α de 5% si la $p-value$ associée à β_0 est inférieure à α .

```

summary(ur.df(rte,type="drift",lag=0))

##
## #####
## # Augmented Dickey-Fuller Test Unit Root Test #
## #####
## 
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.099711 -0.008073  0.000189  0.007924  0.076698
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0005764 0.0003995  1.443   0.149
## z.lag.1     -1.0551305 0.0256477 -41.139 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01551 on 1507 degrees of freedom
## Multiple R-squared:  0.529, Adjusted R-squared:  0.5287
## F-statistic:  1692 on 1 and 1507 DF, p-value: < 2.2e-16
##
##
## Value of test-statistic is: -41.1394 846.2282
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78

```

On observe que la p -value associée à β_0 (intercept) est de $0.149 > 0.05$. Nous acceptons H_0 avec un risque de 5%, le coefficient β_0 n'est pas significatif.

```

summary(ur.df(rte,type="none",lag=0))

##
## #####
## # Augmented Dickey-Fuller Test Unit Root Test #
## #####
## 
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.099050 -0.007504  0.000739  0.008511  0.077271

```

```

## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## z.lag.1 -1.05376   0.02564  -41.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.01552 on 1508 degrees of freedom
## Multiple R-squared:  0.5283, Adjusted R-squared:  0.528 
## F-statistic:  1689 on 1 and 1508 DF,  p-value: < 2.2e-16
## 
## 
## Value of test-statistic is: -41.0995
## 
## Critical values for test statistics:
##      1pct 5pct 10pct
## tau1 -2.58 -1.95 -1.62

```

Nous testons maintenant :

$$H_0 : \rho - 1 = 0 \quad vs \quad H_a : |\rho| < 1$$

On remarque que notre test-statistic associée à $(\rho - 1)$ est de -41.0995, ce qui est inférieur à la valeur critique à 5% qui est de -2.86. On rejette alors H_0 et on conclut que $|\rho| < 1$. **Nous concluons alors que le PGD ayant généré notre série est stationnaire.**

Les conclusions du test Dickey-Fuller ne sont valables que si les ϵ_t ne sont pas autocorrélés. Si les ϵ_t sont autocorrélés, il faudrait corriger l'erreur due à cette autocorrélation en intégrant des variables la prenant en compte, ce qui serait effectué via un test de Dickey-Fuller augmenté. Testons l'autocorrélation avec l'ACF et le PACF des résidus des régressions Dickey-Fuller:

```
plot(ur.df(rte,type= "none",lags=0))
```

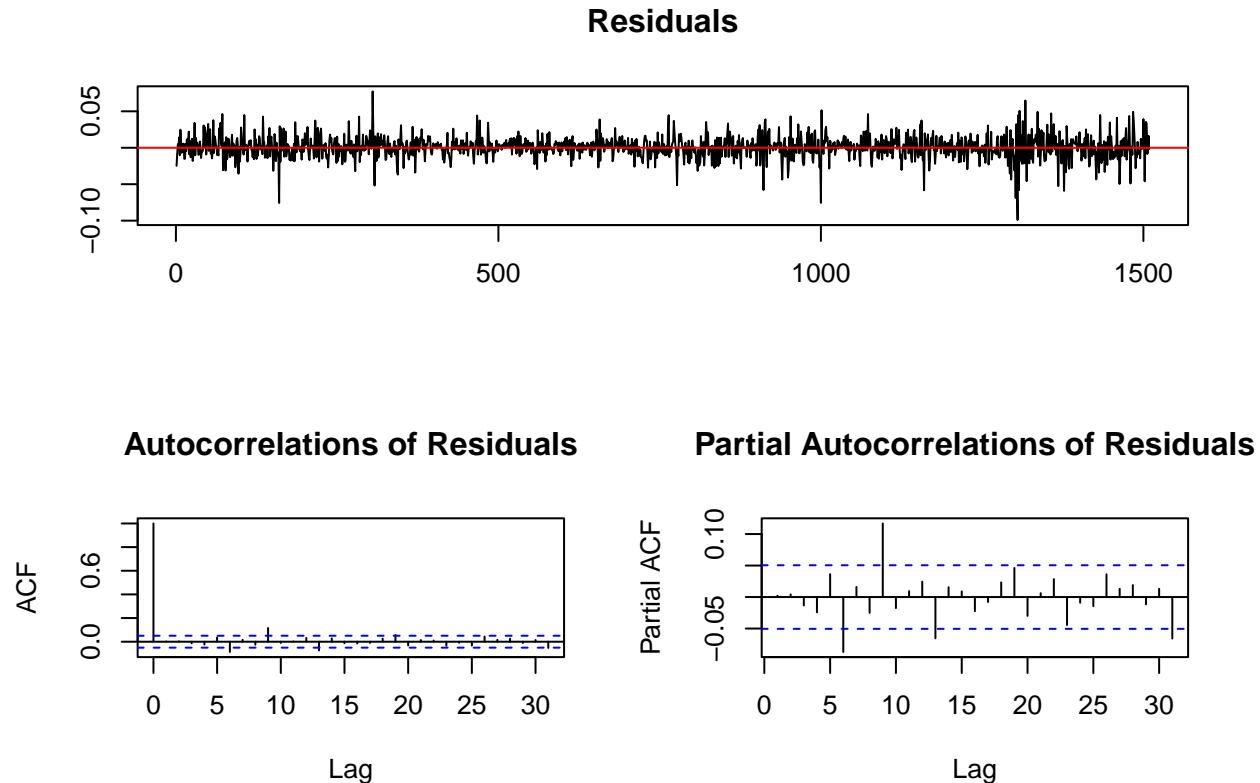


Figure 6: Corrélogramme des résidus de la régression DF *none*

Nous constatons dans le graphique que les aléas sont auto-corrélés aux ordres 6, 9 et 31. Notre conclusion concernant l'absence de racine unitaire dans le rendement logarithmique de notre action n'est pas valide. **Nous devons effectuer un test racine unitaire dans le cadre de la régression ADF.**

1.2) Dickey-Fuller Augmenté (ADF)

Le test de Dickey-Fuller augmenté est un test de Dickey-Fuller qui inclut des variables explicatives en plus, qui est la variable dépendante retardée jusqu'à l'ordre P (P est le nombre de retards que nous ajoutons dans les régressions pour tenir compte de l'autocorrélation dans les aléas). Il est généralement nécessaire de garder la même spécification que celle retenue dans le test de Dickey-Fuller classique pour sa version augmentée. Nous retenons donc la spécification “drift”.

La procédure à mettre en oeuvre débute par le calcul de la valeur maximale de retard à introduire : Pmax. P ne doit être ni trop grand (perte de puissance) ni trop petit (distorsion de la taille du test) pour s'assurer de la justesse de notre test.

La valeur de Pmax est calculée par la formule de Schwert (1989) : $\lfloor 12 \times (T/100)^{0,25} \rfloor$

```
Schwert<-as.integer(12*(T/100)^(0.25))
pmax<-Schwert
pmax
```

```
## [1] 23
```

Nous devons maintenant minimiser le critère d'information de Ng et Perron (2001), le $MAIC(p)$, calculé pour différentes valeurs de p allant de 0 à Pmax (ici de 0 à 26). Le $MAIC(p)$ est donné par la formule

suivante :

$$MAIC(p) = \ln(\hat{\sigma}_p^2) + 2 \frac{(\tau_T(p) + p)}{(T - P_{max})}$$

```
summary(CADFtest(rte, criterion="MAIC", type="none", max.lag.y=pmax))

## Augmented DF test
##                                     ADF test
## t-test statistic:                 -7.548069e+00
## p-value:                         8.471954e-13
## Max lag of the diff. dependent variable: 2.100000e+01
##
## Call:
## dynlm(formula = formula(model), start = obs.1, end = obs.T)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.098497 -0.007491  0.000452  0.008457  0.074633
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## L(y, 1)     -0.95489  0.12651  -7.548 8.47e-13 ***
## L(d(y), 1)  -0.08072  0.12384  -0.652  0.515    
## L(d(y), 2)  -0.07947  0.12110  -0.656  0.512    
## L(d(y), 3)  -0.08791  0.11845  -0.742  0.458    
## L(d(y), 4)  -0.10792  0.11545  -0.935  0.350    
## L(d(y), 5)  -0.07019  0.11222  -0.626  0.532    
## L(d(y), 6)  -0.14339  0.10893  -1.316  0.188    
## L(d(y), 7)  -0.13810  0.10569  -1.307  0.192    
## L(d(y), 8)  -0.15696  0.10220  -1.536  0.125    
## L(d(y), 9)  -0.03810  0.09855  -0.387  0.699    
## L(d(y), 10) -0.05880  0.09518  -0.618  0.537    
## L(d(y), 11) -0.04516  0.09146  -0.494  0.622    
## L(d(y), 12) -0.01929  0.08746  -0.221  0.825    
## L(d(y), 13) -0.08661  0.08342  -1.038  0.299    
## L(d(y), 14) -0.07467  0.07810  -0.956  0.339    
## L(d(y), 15) -0.06247  0.07256  -0.861  0.389    
## L(d(y), 16) -0.07953  0.06652  -1.196  0.232    
## L(d(y), 17) -0.08972  0.06079  -1.476  0.140    
## L(d(y), 18) -0.06364  0.05397  -1.179  0.239    
## L(d(y), 19) -0.01097  0.04663  -0.235  0.814    
## L(d(y), 20) -0.03913  0.03799  -1.030  0.303    
## L(d(y), 21) -0.03391  0.02633  -1.288  0.198    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01539 on 1464 degrees of freedom
## Multiple R-squared:  0.5416, Adjusted R-squared:  0.5348
## F-statistic: 2.549 on 21 and 1464 DF,  p-value: 0.0001413
```

“Max lag of the diff. dependent variable” indique le nombre de variables explicatives à ajouter pour tenir compte de l'autocorrélation. Nous garderons alors 2 variable explicative dans notre test de Dickey-Fuller Augmenté.

Passons maintenant à la réalisation du test :

```

summary(ur.df(rte, type = "none", lag = 2))

##
## #####
## # Augmented Dickey-Fuller Test Unit Root Test #
## #####
## 
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.099737 -0.007463  0.000757  0.008561  0.077395
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -1.054723  0.046057 -22.900 <2e-16 ***
## z.diff.lag1  0.003831  0.037371   0.103   0.918
## z.diff.lag2  0.009590  0.025695   0.373   0.709
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01552 on 1504 degrees of freedom
## Multiple R-squared:  0.5257, Adjusted R-squared:  0.5248
## F-statistic: 555.7 on 3 and 1504 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -22.9004
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62

```

Nous observons que la test-statistic associée à la variable ajoutée par rapport au test de DickeyFuller classique est -32.2875. Or on a $-32.2875 < -2.86$. Nous rejetons donc H_0 . **Le PGD qui a généré notre série est stationnaire.**

Approche “top-down” Nous allons employer une procédure “Top-Down” pour vérifier la robustesse de notre résultat. Pour cela, nous allons employer $lags = pmax$ (Schwert) et diminuer la valeur de $lags$ jusqu'à ce que le dernier γ ait une valeur de la t – statistique calculée supérieure en valeur absolue à 1,6.

```

summary(ur.df(rte,type= "none",lags=pmax-11))

##
## #####
## # Augmented Dickey-Fuller Test Unit Root Test #
## #####
## 
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##      Min       1Q   Median       3Q      Max
## -0.099737 -0.007463  0.000757  0.008561  0.077395
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -1.054723  0.046057 -22.900 <2e-16 ***
## z.diff.lag1  0.003831  0.037371   0.103   0.918
## z.diff.lag2  0.009590  0.025695   0.373   0.709
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01552 on 1504 degrees of freedom
## Multiple R-squared:  0.5257, Adjusted R-squared:  0.5248
## F-statistic: 555.7 on 3 and 1504 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -22.9004
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62

```

```

## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##   Min     1Q Median     3Q    Max
## -0.098766 -0.007398  0.000532  0.008556  0.075150
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1      -1.040968  0.097788 -10.645 <2e-16 ***
## z.diff.lag1    0.003651  0.094475  0.039   0.969
## z.diff.lag2    0.003161  0.090796  0.035   0.972
## z.diff.lag3   -0.002684  0.086917 -0.031   0.975
## z.diff.lag4   -0.021984  0.082941 -0.265   0.791
## z.diff.lag5    0.016159  0.077627  0.208   0.835
## z.diff.lag6   -0.062408  0.072119 -0.865   0.387
## z.diff.lag7   -0.056430  0.066063 -0.854   0.393
## z.diff.lag8   -0.076944  0.060415 -1.274   0.203
## z.diff.lag9    0.040056  0.053693  0.746   0.456
## z.diff.lag10   0.029307  0.046348  0.632   0.527
## z.diff.lag11   0.037882  0.037674  1.006   0.315
## z.diff.lag12   0.065452  0.026018  2.516   0.012 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01537 on 1484 degrees of freedom
## Multiple R-squared:  0.5391, Adjusted R-squared:  0.535
## F-statistic: 133.5 on 13 and 1484 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -10.6452
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62

```

Il nous faut descendre à lags=12 pour que le dernier γ ait une valeur de la t-statistique calculée supérieure en valeur absolue à 1,6. La p -value associée à β_0 est de $0.012 < 0.05$, donc β_0 est significatif. Nous observons que la test-statistic est -9.9296. Or on a $-9.9296 < -2.86$. Nous rejetons donc H_0 . Le PGD qui a généré notre série est stationnaire.

Cependant Perron a démontré que les résultats des tests DF et ADF ne sont valables que s'il n'y a pas de changements conjoncturels ou structurels dans les données.

1.3) Zivot-Andrews (ZA)

La dynamique d'un processus économique particulier peut être affectée par des changements structurels qui engendrent une instabilité dans le temps de cette dynamique. Les changements structurels sont dus aux crises, aux changements législatifs, institutionnels, technologiques, à une redéfinition des séries de données. Cette instabilité peut toucher le niveau de la série, sa variance, ses autocorrélations ou une combinaison des 3.

La formalisation des changements structurels est complexe car elle dépend de plusieurs facteurs :

- le nombre de changements structurels : combien de points de rupture ?
- leur date d'occurrence : ces dates sont-elles exogènes (connues a posteriori) ou endogènes ?

- la manière dont le changement survient : abruptement (AO="Additive Outliers") ou graduellement (IO="Innovational Outliers")
- de ce qui est touché (niveau de la série, le taux de croissance de la série ou bien les deux).

Dans la suite nous allons introduire des variables pour modéliser ces changements structurels. Soit T_B la date de changement structurel, DU_t la variable à ajouter pour modéliser un changement dans le niveau de la partie déterministe de la série et DT_t la variable à ajouter pour modéliser un changement dans la pente de la partie déterministe de la série.

$$DU_t = \begin{cases} 1 & \text{si } t \geq T_B + 1 \\ 0 & \text{sinon.} \end{cases}$$

$$DT_t = \begin{cases} t - T_B & \text{si } t \geq T_B + 1 \\ 0 & \text{sinon.} \end{cases}$$

Deux spécifications sont possibles :

- La spécification “crash” :

$$y_t = \beta_0 + \beta_1 t + \rho y_{t-1} + \delta DU_t(T_B) + \sum_{j=1}^p \gamma_j \Delta y_{t-j} + \epsilon_t$$

Le niveau de la série est touché ici.

- La spécification “both” :

$$y_t = \beta_0 + \beta_1 t + \rho y_{t-1} + \delta_1 DU_t(T_B) + \delta_2 DT_t(T_B) + \sum_{j=1}^p \gamma_j \Delta y_{t-j} + \epsilon_t$$

Le niveau de la série et son taux de croissance sont touchés ici. Cette spécification est plus complète, il faudra donc commencer par celle-ci.

Les hypothèses à tester sont alors les suivantes:

$$H_0 : \rho = 1 \quad vs \quad H_a : |\rho| < 1$$

La règle de décision est de rejeter H_0 si la statistique calculée est inférieure à la valeur critique au seuil de risque α de 5%.

Notons que pour la valeur de lag, qui nous permet de modéliser l'autocorrélation, nous pouvons choisir lags=1 d'après la méthode “Top-Down” vu précédemment, lags=1 d'après le critère du BIC (détailé en annexe) ou encore utiliser une méthode décrite par Perron (1989) qui consiste à partir de la formule de Schwert puis de diminuer le nombre de γ de tel sorte que le dernier ait une statistique t en valeur absolue < 1.6 et l'avant dernier une statistique t en valeur absolue > 1.6 .

- Spécification “both”

```
summary(ur.za(rte, model="both", lag=2))
```

```
## #####
## # Zivot-Andrews Unit Root Test #
## #####
##
## Call:
## lm(formula = testmat)
##
```

```

## Residuals:
##      Min       1Q    Median       3Q      Max
## -0.100338 -0.007973  0.000104  0.007990  0.076699
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.494e-03 3.982e-03 -0.375  0.7076
## y.l1        -7.493e-02 4.632e-02 -1.618  0.1060
## trend        1.659e-04 9.178e-05  1.808  0.0708 .
## y.dl1        1.747e-02 3.749e-02  0.466  0.6414
## y.dl2        1.614e-02 2.570e-02  0.628  0.5302
## du          -9.661e-03 3.766e-03 -2.565  0.0104 *
## dt          -1.667e-04 9.179e-05 -1.816  0.0696 .
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01548 on 1500 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.009132, Adjusted R-squared:  0.005168
## F-statistic: 2.304 on 6 and 1500 DF, p-value: 0.03221
##
##
## Teststatistic: -23.2043
## Critical values: 0.01= -5.57 0.05= -5.08 0.1= -4.82
##
## Potential break point at position: 73

```

Nous constatons que la statistique t en valeur absolue de γ_2 est bien inférieure à 1,6 en valeur absolue ($|0.628| < 1.6$). La p -value de δ_2 est de $0.0696 > 0.05$ donc δ_2 n'est pas significatif.

- Spécification “intercept”

```

summary(ur.za(rte, model="intercept",lag=2))

##
## #####
## # Zivot-Andrews Unit Root Test #
## #####
##
##
## Call:
## lm(formula = testmat)
##
## Residuals:
##      Min       1Q    Median       3Q      Max
## -0.099799 -0.007777  0.000117  0.008043  0.076138
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.864e-03 8.083e-04  2.306  0.0212 *
## y.l1        -7.091e-02 4.628e-02 -1.532  0.1257
## trend       -1.769e-06 9.329e-07 -1.896  0.0581 .
## y.dl1        1.435e-02 3.746e-02  0.383  0.7017
## y.dl2        1.475e-02 2.570e-02  0.574  0.5661
## du          8.963e-03 4.228e-03  2.120  0.0342 *
##
## ---
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01549 on 1501 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.007495, Adjusted R-squared:  0.004189
## F-statistic: 2.267 on 5 and 1501 DF, p-value: 0.04568
##
##
## Teststatistic: -23.1416
## Critical values: 0.01= -5.34 0.05= -4.8 0.1= -4.58
##
## Potential break point at position: 1496

```

Nous constatons que la statistique t en valeur absolue de γ_2 est bien inférieure à 1,6 en valeur absolue ($|-0.441| < 1.6$) et la statistique t en valeur absolue de γ_{25} est bien supérieure à 1,6 en valeur absolue ($|1.609| > 1.6$). La $p - value$ de δ_1 est de $0.0104 < 0.05$ donc δ_1 est significatif. La $p - value$ de γ_1 est de $0.02134 < 0.05$ donc γ_1 est significatif. La statistique de test est $-10.3525 < -4.8$ la valeur critique à 5% . On rejette H_0 et ρ est différent de 1. Nous avons un PGD stationnaire avec une rupture structurelle En ce qui concerne la date de rupture, nous avons “Potential break point at position: 2087”. La date de rupture est donc le 10 décembre 2020 :

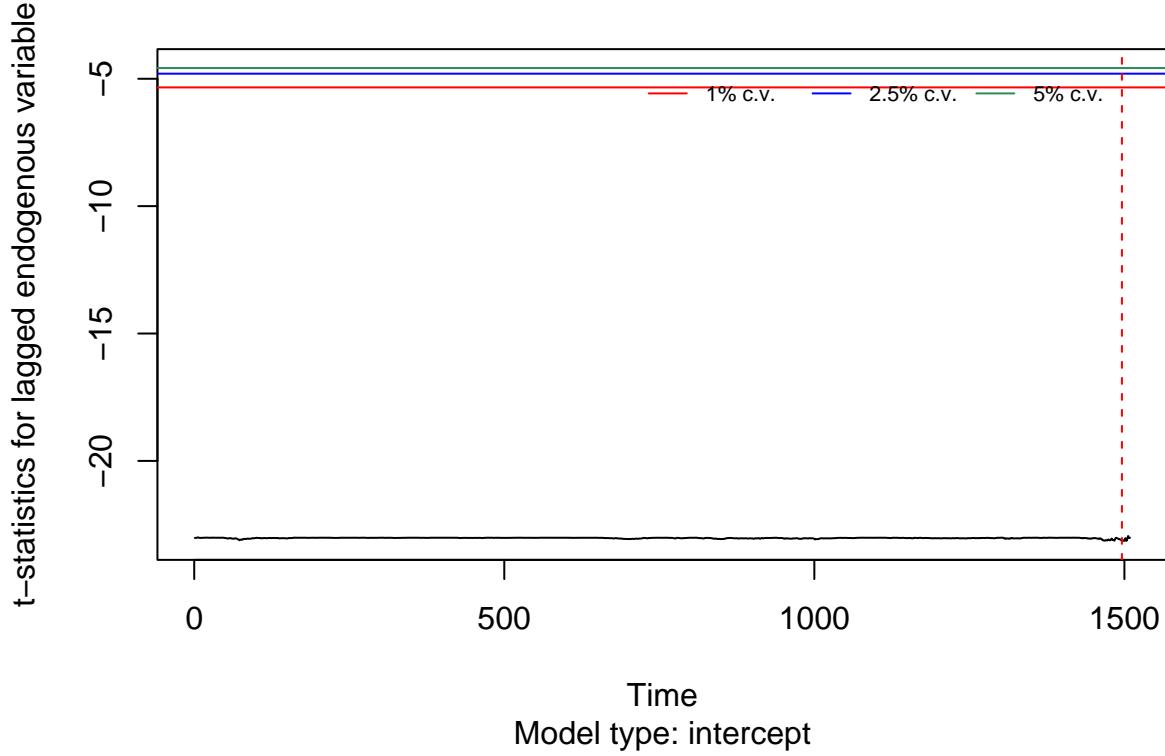
```
dates[1496]
```

```

## [1] "2020-12-10"
plot(ur.za(rte, model="intercept", lag=2))

```

Zivot and Andrews Unit Root Test



observons bien graphiquement une rupture le 5 avril 2018 (à la 2077 ème valeur).

Nous

1.4) Lee et Strazicich (LS)

Le test de Lee et Strazicich est une généralisation du test de racines unitaires de Schmidt et Phillips. Cette généralisation emploie la classification des changements structurels de Perron (1989) "crash" et "both" sachant qu'ils les appellent respectivement "crash" et "break" et en introduisant 2 dates de rupture endogènes. Le modèle est le suivant :

$$y_t = \delta' Z_t + e_y$$

$$e_t = \beta e_{t-1} + \epsilon_t$$

avec $\epsilon \sim N(0, \sigma^2)$ et Z la matrice des variables exogènes.

Soit T_{B1} et T_{B2} respectivement les dates du premier et deuxième changements structurels:

- "crash" : $Z_t = [\iota, tendance, DU_{1t}, DU_{2t}]'$ avec $DU_{jt} = 1$ si $t \geq T_{Bj} + 1$ pour $j=1,2$ et 0 sinon
- "break": $Z_t = [\iota, tendance, DU_{1t}, DU_{2t}, DT_{1t}, DT_{2t}]'$ avec $DT_{jt} = t - T_{Bj}$ si $t \geq T_{Bj} + 1$ pour $j=1,2$ et 0 sinon

Avec la spécification "crash", nous testons :

$$H_0 : y_t = \mu_0 + d_1 B_{1t} + d_2 B_{2t} y_{t-1} + v_{1t} \quad vs \quad H_a : y_t = \mu_1 + \gamma trend_t + d_1 D_{1t} + d_2 D_{2t} + v_{2t}$$

On rejette H_0 si la valeur critique au croisement du λ estimé et du seuil de risque 5% est supérieure à la statistique calculée. Dans tous les cas, le PGD comprend des changements structurels sous H_0 et sous H_a .

D'après le test de Zivot et Andrews, δ_2 n'est pas significatif, nous utilisons la spécification "crash" et lags=22.

Le test LS peut s'effectuer avec ou sans bootstrap. Dans notre cas, nous n'utiliserons celui pas avec bootstrap car celui-ci est employé lorsque nous avons un faible nombre d'observation. Ici nous en avons 2263 observations ce qui est largement supérieur à 100 donc bootstrap n'est pas utile.

```
LS_test <- ur.ls(y=rte, model = "crash", breaks = 1, lags = 2, method="GTOS", pn = 0.1, print.results = TRUE)
## [1] -12.45815
## [1] "First possible structural break at position: 844"
## [1] "The location of the first break - lambda_1: 0.6 , with the number of total observations: 1510"
## Critical values - Crash model:
##          1%      5%     10%
## [1,] -4.239 -3.566 -3.211
## [1] "Number of lags determined by general-to-specific lag selection: 2"
## Runtime:
## Time difference of 0.04159474 mins
```

Nous avons besoin d'introduire 2 variables explicatives additionnelles dans notre modèle pour prendre en compte l'autocorrélation. La valeur de la statistique de test est $-12.46 < -3.566$ (valeur critique à 5%), nous rejetons donc H_0 et il n'y a pas de racine unitaire.

```
dates[844]
```

```
## [1] "2018-05-10"
```

Nous concluons que le PGD qui a générée notre série rte est stationnaire.

III-Conclusion

On a montré que :

- notre série rte est asymétrique (propriété 1)
- la distribution de rte est leptokurtique (propriété 2)
- notre série rte présente des clusters de volatilité (propriété 4)

- Les queues de distribution des aléas de notre ARMA-GARCH sont plus épaisses que celles d'une loi normale (propriété 5)
- notre série rte présente un effet de levier (propriété 6)
- le PGD de notre série rte est stationnaire (propriété 8)
- La série rte^2 présente de fortes auto-corrélations alors que notre série rte ne présente pas d'autocorrélation faible (propriété 3)
- notre série rte a de la saisonnalité (propriété 7)

IV-Annexes

Nous allons maintenant vérifier les 8 propriétés sur la série rtt .

```
plot(dates[1511:N], rtt, type='l', col=2, xlab='Années')
```

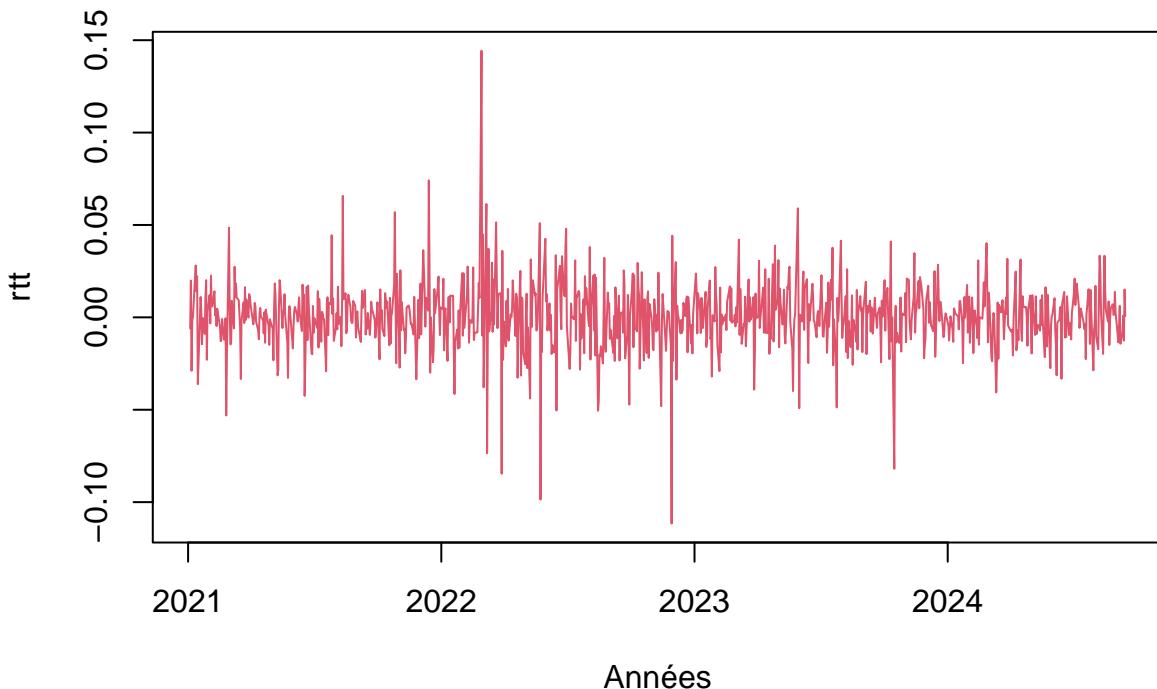


Figure 7: Chronogramme de notre série rtt

Calculons la moyenne de rtt

```
T=length(rtt)
rttbar<-mean(rtt) #moyenne empirique
rttbar
```

```
## [1] 0.000488238
stt=sd(rtt) #écart type estimé
```

Nous regardons si notre moyenne empirique, proche de 0, est statistiquement nulle :

```
t.test(rtt)
```

```
##
## One Sample t-test
##
```

```

## data: rtt
## t = 0.81847, df = 929, p-value = 0.4133
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.0006824615 0.0016589376
## sample estimates:
## mean of x
## 0.000488238

```

La p -value vaut $0.4133 > 0.05$ donc on accepte H_0 et la moyenne de rtt est nulle.

1) Propriété 1: Asymétrie Perte/Gain

```
agostino.test(rtt)
```

```

##
## D'Agostino skewness test
##
## data: rtt
## skew = 0.019528, z = 0.245023, p-value = 0.8064
## alternative hypothesis: data have a skewness

```

On trouve une p -value de $p\text{-value} = 0.8064 > 0.05$, on ne rejette donc pas H_0 et on conclut donc que la distribution est symétrique.

2) Propriété 2 : Queues de distribution épaissees

```
anscombe.test(rtt)
```

```

##
## Anscombe-Glynn kurtosis test
##
## data: rtt
## kurt = 10.887, z = 12.291, p-value < 2.2e-16
## alternative hypothesis: kurtosis is not equal to 3

```

On trouve une p -value < 0.05 donc on rejette H_0 et de plus, $kurt = 10.036$, donc la distribution est leptokurtique. Les queues de distribution de rtt sont plus épaissees que celles d'une loi normale.

3) Propriété 3 : Autocorrélations des carrés des rendements fortes et faibles pour les rendements

```

op<-par(mfrow=c(2,2))
Acf(rtt,main='ACF du rendement logarithmique')
Pacf(rtt,main="PACF du rendement logarithmique")
Acf(rtt^2,main='ACF du rendement logarithmique au carré')
Pacf(rtt^2,main="PACF du rendement logarithmique au carré")

```

```
par(op)
```

On remarque la présence de faibles autocorrélations des carrés des rendements et de faibles autocorrélations des rendements.

```

pvaluesrtt =rep(0,30)
pvaluesrtt2 =rep(0,20)
for (i in 1:30 ) {

```

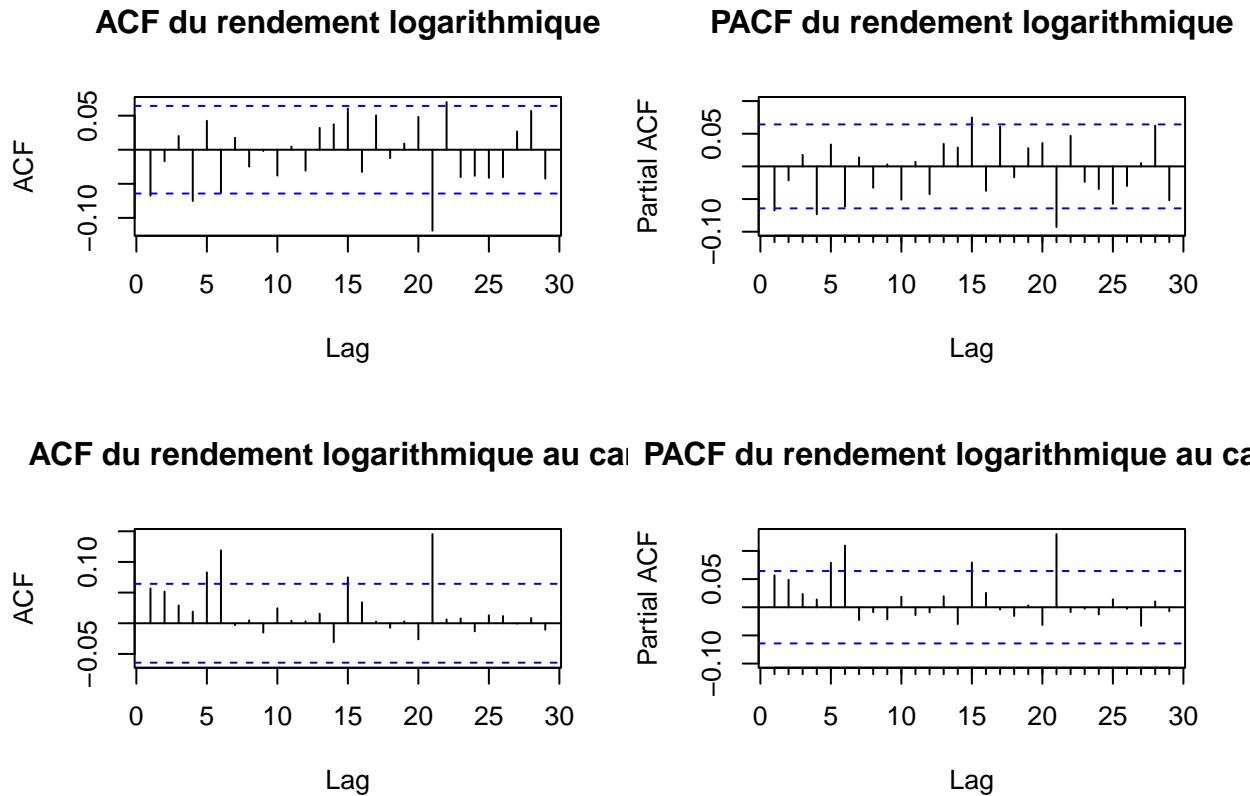


Figure 8: Corrélogrammes de rtt et rtt^2

```

pvaluesrtt[i] = Box.test(rtt,lag=i,type="Ljung-Box")$p.value
pvaluesrtt2[i] = Box.test(rtt^2,lag=i,type="Ljung-Box")$p.value
}

pvaluesrtt

## [1] 0.0395852094 0.1055753558 0.1810587271 0.0377898967 0.0368422507
## [6] 0.0164759727 0.0267746691 0.0370015663 0.0589298567 0.0593000781
## [11] 0.0870649938 0.0971981367 0.1049273810 0.1030608375 0.0583318382
## [16] 0.0627287089 0.0469908420 0.0624768676 0.0824727095 0.0657495690
## [21] 0.0025563304 0.0009740225 0.0009519892 0.0009676349 0.0009166385
## [26] 0.0008863796 0.0010966086 0.0006713280 0.0006136408 0.0009141067

```

Les $p-value$ du test de LB sur les rendements sont toutes inférieures à 0.05 donc on rejette H_0 et donc il y a de l'autocorrélation dans les rendements logarithmiques au carré.

```

regrtt<-Arima(rtt, order=c(0,0,20))
coeftest(regrtt)

```

```

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ma1      -0.05079189  0.03379153 -1.5031  0.13281
## ma2      -0.03688832  0.03417126 -1.0795  0.28036
## ma3       0.03121383  0.03451447  0.9044  0.36580
## ma4      -0.07439746  0.03436167 -2.1651  0.03038 *

```

```

## ma5      0.04587202  0.03509956  1.3069  0.19124
## ma6     -0.04833609  0.03531265 -1.3688  0.17106
## ma7      0.02662714  0.03495757  0.7617  0.44624
## ma8     -0.02340503  0.03514798 -0.6659  0.50548
## ma9      0.00946727  0.03526900  0.2684  0.78837
## ma10    -0.02777774  0.03478118 -0.7986  0.42450
## ma11     0.00890419  0.03273380  0.2720  0.78561
## ma12    -0.01431695  0.03779959 -0.3788  0.70487
## ma13     0.02493929  0.03771796  0.6612  0.50848
## ma14     0.05025049  0.03490086  1.4398  0.14992
## ma15     0.05833522  0.03654267  1.5964  0.11041
## ma16    -0.01143006  0.03912700 -0.2921  0.77019
## ma17     0.03075171  0.03777593  0.8141  0.41561
## ma18     0.02194476  0.03835089  0.5722  0.56718
## ma19     0.00221314  0.03914416  0.0565  0.95491
## ma20     0.04831239  0.03756472  1.2861  0.19840
## intercept 0.00049062  0.00062834  0.7808  0.43491
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Tous nos coefficients ne sont pas significatifs (intercept,ma6,ma5, ma3, ma2), on va donc les enlever :

```

regrtt<-Arima(rtt, order=c(0,0,20),fixed=c(0,0,0,NA,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0))
coeftest(regrtt)

```

```

##
## z test of coefficients:
##
##          Estimate Std. Error z value Pr(>|z|)
## ma4   -0.081214  0.033638 -2.4143  0.01576 *
## ma15   0.064203  0.032727  1.9618  0.04979 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Tous nos coefficients sont maintenant significatifs. Notre modèle MA(4) modélise l'autocorrélation de rtt , vérifions si ses aléas ont une espérance nulle et ne sont pas autocorrélés. Commençons par tester :

$$H_0 : E(\epsilon) = 0 \quad vs \quad H_a : E(\epsilon) \neq 0$$

```

residurtt<-regrtt$res
t.test(residurtt)

##
##  One Sample t-test
##
## data:  residurtt
## t = 0.83582, df = 929, p-value = 0.4035
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0006687157  0.0016608524
## sample estimates:
##  mean of x
##  0.0004960683

```

La p -value est de $0.373 > 0.05$, on accepte H_0 et donc l'espérance des aléas est nulle.

Testons maintenant : H_0 :Absence d'autocorrélation jusqu'à l'ordre K vs H_a Présence d'autocorrélation

```

residuvrtt=(residurtt-mean(residurtt))/sd(residurtt)
K<-20
tmp<-rep(0,K)
for(i in 1:K){
tmp[i]<-Box.test(residuvrtt,lag=i,type="Ljung-Box")$p.value
}
tmp

## [1] 0.05158174 0.10143814 0.17760181 0.29506777 0.28334307 0.13560545
## [7] 0.18636898 0.21909933 0.28959746 0.26674181 0.33067535 0.30930725
## [13] 0.30507655 0.27637610 0.34130956 0.34780501 0.28252007 0.34147193
## [19] 0.40106532 0.35968335

```

Les $p - value$ sont toutes supérieure à 0,05, on ne rejette pas H_0 . Il n'y a pas d'autocorrélation jusqu'à l'ordre 20 Nous choisissons un MA(2) pour modéliser l'autocorrélation dans rtt .

4) Propriété 4 : Clusters de volatilité

```

LM1<-ArchTest(as.numeric(rtt),lag=1)
LM1

```

```

##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
##  data:  as.numeric(rtt)
##  Chi-squared = 2.9976, df = 1, p-value = 0.08339

```

La $p - value > 0.05$, on accepte donc H_0 et donc il y a de l'homoscédasticité conditionnelle

```

LM2<-ArchTest(as.numeric(rtt),lag=2)
LM2

```

```

##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
##  data:  as.numeric(rtt)
##  Chi-squared = 5.1912, df = 2, p-value = 0.0746

```

La $p - value > 0.05$, on accepte donc H_0 et donc il y a de l'homoscédasticité conditionnelle à l'ordre 2.

```

LM20<-ArchTest(as.numeric(rtt),lag=20)
LM20

```

```

##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
##  data:  as.numeric(rtt)
##  Chi-squared = 32.441, df = 20, p-value = 0.03882

```

La $p - value < 0.05$, on rejette donc H_0 et donc il y a présence de clusters de volatilité à l'ordre 20 dans rtt .

```

LM40<-ArchTest(as.numeric(rtt),lag=40)
LM40

```

```

##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
##  data:  as.numeric(rtt)
##  Chi-squared = 55.933, df = 40, p-value = 0.04841

```

La p -value <0.05, on rejette donc H_0 et donc il y a présence de clusters de volatilité dans rtt .

5) Propriété 5 : Queues épaissees conditionnelles

```

volat=garch(residuvrtt,order=c(1,1))

##  

## ***** ESTIMATION WITH ANALYTICAL GRADIENT *****  

##  

##  

##      I      INITIAL X(I)          D(I)  

##  

##      1      9.000000e-01      1.000e+00  

##      2      5.000000e-02      1.000e+00  

##      3      5.000000e-02      1.000e+00  

##  

##      IT      NF      F          RELDF      PRELDF      RELDX      STPPAR      D*STEP      NPRELDF  

##      0      1  4.568e+02  

##      1      4  4.561e+02  1.57e-03  1.41e-02  5.3e-02  7.0e+02  9.6e-02  4.94e+00  

##      2      6  4.550e+02  2.34e-03  2.74e-03  1.5e-02  2.5e+00  3.9e-02  8.62e-01  

##      3      7  4.546e+02  9.19e-04  1.43e-03  1.7e-02  3.5e+02  3.9e-02  3.52e-01  

##      4      9  4.541e+02  1.13e-03  1.75e-03  4.2e-02  3.4e+00  8.2e-02  2.39e-01  

##      5     10  4.528e+02  2.90e-03  3.90e-03  8.7e-02  1.9e+00  1.6e-01  6.46e-02  

##      6     11  4.521e+02  1.59e-03  2.96e-03  1.1e-01  1.7e+00  1.6e-01  1.89e-02  

##      7     13  4.507e+02  2.95e-03  2.50e-03  1.3e-01  0.0e+00  1.6e-01  4.23e-03  

##      8     14  4.497e+02  2.24e-03  4.54e-03  2.2e-01  1.1e+00  3.3e-01  1.22e-02  

##      9     17  4.495e+02  4.11e-04  7.78e-04  4.9e-03  6.6e+00  9.2e-03  3.58e-03  

##     10     18  4.493e+02  5.30e-04  6.45e-04  6.7e-03  1.8e+00  9.2e-03  2.29e-03  

##     11     19  4.492e+02  2.42e-04  2.78e-04  5.5e-03  1.6e+00  9.2e-03  7.24e-04  

##     12     21  4.491e+02  1.52e-04  2.28e-04  2.1e-02  4.1e-01  3.8e-02  2.44e-04  

##     13     22  4.491e+02  1.57e-05  2.16e-05  2.2e-03  0.0e+00  4.5e-03  2.16e-05  

##     14     24  4.491e+02  4.11e-06  1.17e-06  2.5e-03  0.0e+00  4.7e-03  1.17e-06  

##     15     25  4.491e+02  3.38e-06  4.14e-07  2.7e-03  0.0e+00  5.1e-03  4.14e-07  

##     16     26  4.491e+02  1.30e-06  6.83e-08  1.1e-03  0.0e+00  2.1e-03  6.83e-08  

##     17     27  4.491e+02  1.84e-07  2.95e-09  1.6e-04  0.0e+00  3.1e-04  2.95e-09  

##     18     28  4.491e+02 -4.07e-09  1.36e-11  5.2e-06  0.0e+00  9.5e-06  1.36e-11  

##  

## ***** RELATIVE FUNCTION CONVERGENCE *****  

##  

##      FUNCTION      4.490966e+02      RELDX      5.193e-06  

##      FUNC. EVALS      28      GRAD. EVALS      18  

##      PRELDF      1.360e-11      NPRELDF      1.360e-11  

##  

##      I      FINAL X(I)          D(I)          G(I)  

##  

##      1      1.992011e-01      1.000e+00      -7.625e-06  

##      2      7.376519e-02      1.000e+00      -5.081e-03  

##      3      7.280281e-01      1.000e+00      -8.024e-04  

summary(volat)

##  

## Call:  

## garch(x = residuvrtt, order = c(1, 1))  

##

```

```

## Model:
## GARCH(1,1)
##
## Residuals:
##      Min       1Q   Median      3Q     Max
## -6.585615 -0.542318 -0.002921  0.556544  8.571779
##
## Coefficient(s):
##    Estimate Std. Error t value Pr(>|t|)
## a0    0.19920    0.05109   3.899 9.66e-05 ***
## a1    0.07377    0.01627   4.535 5.76e-06 ***
## b1    0.72803    0.06154  11.829 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
## Jarque Bera Test
##
## data: Residuals
## X-squared = 3289.4, df = 2, p-value < 2.2e-16
##
##
## Box-Ljung test
##
## data: Squared.Residuals
## X-squared = 0.021776, df = 1, p-value = 0.8827
ArchTest(volat$residuals,lag=1)

##
## ARCH LM-test; Null hypothesis: no ARCH effects
##
## data: volat$residuals
## Chi-squared = 0.021685, df = 1, p-value = 0.8829

```

Nous avons une $p - value$ de $0.8829 > 0.05$ donc le modèle Garch a pris en compte les effets Arch et le modèle Garch permet de modéliser notre MA(20) tout en prenant en compte l'hétéroscédascticité. Nous regardons maintenant si les queues de distribution des aléas de notre ARMA-GARCH sont plus épaisses que celles d'une loi normale. Pour cela nous testons : $H_0 : kurtosis = 3$ vs $H_a : kurtosis \neq 3$ On rejette H_0 si la $p - value$ est inférieure à 0.05.

```

anscombe.test(volat$res)

##
## Anscombe-Glynn kurtosis test
##
## data: volat$res
## kurt = 12.208, z = 12.879, p-value < 2.2e-16
## alternative hypothesis: kurtosis is not equal to 3

```

On trouve une $p - value < 2.2e-16$ donc on rejette H_0 . Ainsi les queues de distribution des aléas de notre ARMA-GARCH sont plus épaisses que celles d'une loi normale.

6) Propriété 6 : Effet de levier

```

sig<-rep(0,length(rtt))
for(t in 1:length(rtt))
{
  sig[t]<-sqrt(sum(rtt[t-22]-(sum(rtt[t-22]/22))^2/22))
}
sigma=sig[24:length(rtt)]*100
plot(log(pt[24:length(rtt)]),type='l',col=2,axes=F,xlab="", ylab="",lwd=3)
axis(2,at=seq(3.5,5,by=0.25))#axe de gauche avec 3.5 += min logpt et 6.5 +=max logpt
par(new=T)
plot(sigma, col="grey",type='l',axes = F,xlab="", ylab="")
axis(4,at=seq(0,2.5,by=0.25))#axe de droite
legend("topleft", c("log(pt)","sigma"),col = c(2, 1),lty=c(1,1))
abline(v = 780,col = 2,lwd = 4)
abline(v = 840,col = 2,lwd = 4)

```

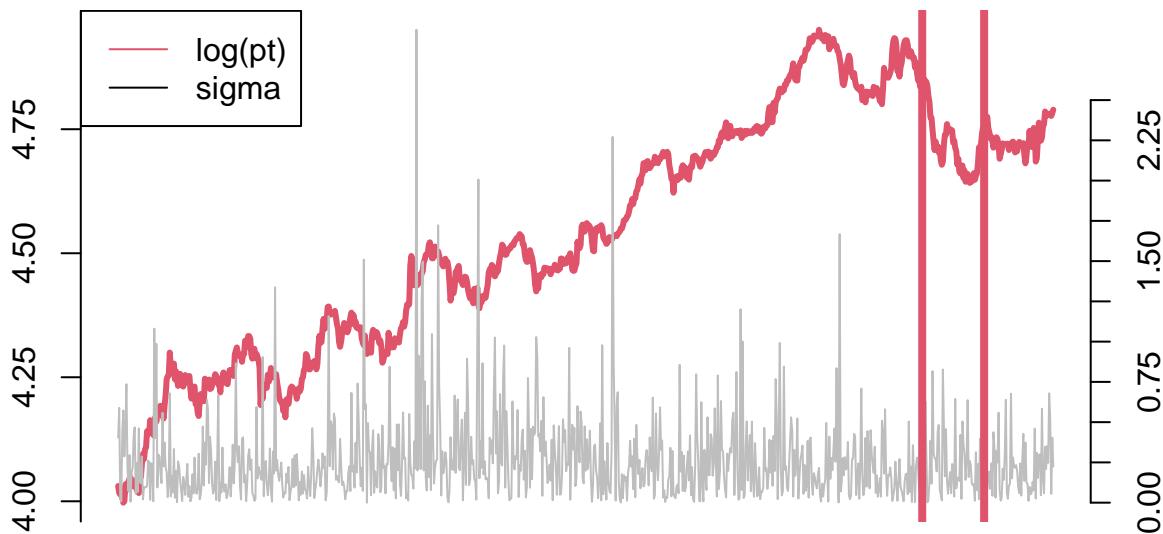


Figure 9: Logarithme de l'action ESLT journalier et écart-type récursif journalier de notre action à partir de janvier 2010

Graphiquement on remarque que pendant une chute de marché (entre les traits rouges) la volatilité n'a pas augmenté. Il n'y a pas d'effet de levier.

7) Propriété 7 : La saisonnalité

- Effet week-end :

```

jour=format(dates[1:T], format = "%A")
tableaures <- data.frame(matrix(NA,ncol=5,nrow=4))
colnames(tableaures) <- c("lundi","mardi","mercredi","jeudi","vendredi")
rownames(tableaures) <- c("moyenne en %","ecart-type annuel en %","skewness","kurtosis")

m<-seq(from=1,to=T,by=5)
rtlun<-as.numeric(rte[m])
lundi<-mean(rtlun)
tableaures[1,1] <- lundi*100
tableaures[2,1] <- sd(rtlun)*100*sqrt(252)

```

```

tableaures[3,1] <- skewness(rtlun)
tableaures[4,1] <- kurtosis(rtlun)

m<-seq(from=2,to=T,by=5)
rtmar<-as.numeric(rte[m])
mardi<-mean(rtmar) #moyenne journaliere
tableaures[1,2] <- mardi*100 #moyenne journaliere en %
tableaures[2,2] <- sd(rtmar)*100*sqrt(252) #ecart-type annualise en %
tableaures[3,2] <- skewness(rtmar)
tableaures[4,2] <- kurtosis(rtmar)

m<-seq(from=3,to=T,by=5)
rtmer<-as.numeric(rte[m])
mer<-mean(rtmer)
tableaures[1,3] <- mer*100
tableaures[2,3] <- sd(rtmer)*100*sqrt(252)
tableaures[3,3] <- skewness(rtmer)
tableaures[4,3] <- kurtosis(rtmer)

m<-seq(from=4,to=T,by=5)
rtjeu<-as.numeric(rte[m])
jeudi<-mean(rtjeu)
tableaures[1,4] <- jeudi*100
tableaures[2,4] <- sd(rtjeu)*100*sqrt(252)
tableaures[3,4] <- skewness(rtjeu)
tableaures[4,4] <- kurtosis(rtjeu)

m<-seq(from=5,to=T,by=5)
rtven<-as.numeric(rte[m])
ven<-mean(rtven)
tableaures[1,5] <- ven*100
tableaures[2,5] <- sd(rtven)*100*sqrt(252)
tableaures[3,5] <- skewness(rtven)
tableaures[4,5] <- kurtosis(rtven)

tableaures

##                               lundi      mardi   mercredi    jeudi
## moyenne en %           0.2376347 -0.096084258  0.03375902  0.07111186
## ecart-type annuel en % 23.2513272 23.255045491 21.88778582 21.21440060
## skewness                 0.2116309 -0.004725557  0.19839095 -0.18997278
## kurtosis                  7.0242814  1.089676600  1.67969563  1.28226330
##                               vendredi
## moyenne en %            0.1979749
## ecart-type annuel en % 19.3758466
## skewness                  0.4900118
## kurtosis                   1.5853658

```

On remarque que l'écart-type le plus fort est celui du lundi. Et les écarts-types diminuent progressivement jusqu'au vendredi. On peut donc attester la présence d'un effet week-end tel qu'énoncé par French et Roll (1986). - Effet janvier :

```

monthplot(rtt, ylab="rendement",main="", cex.main=1,col.base=2,lwd.base=3,col='grey',lwd=0.5)
abline(h = rttbar,col = 3,lwd = 1)

```

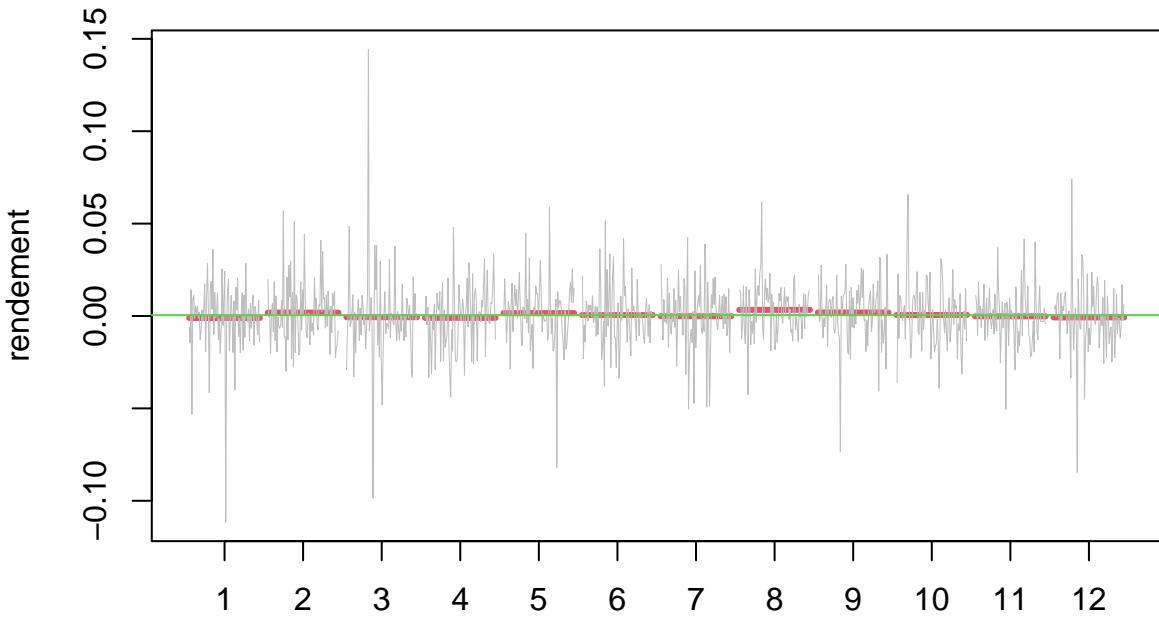


Figure 10: Rendement logarithmique par mois de l'action ESLT à partir de janvier 2010

Les mois de Janvier, Avril et Décembre n'ont pas les moyennes les plus hautes, nous ne pouvons conclure la présence d'effet Janvier.

8) Propriété 8 : Stationnarité

- Dickey-Fuller :

```
summary(ur.df(rtt,type="trend",lag=0))

##
## #####
## # Augmented Dickey-Fuller Test Unit Root Test #
## #####
## 
## Test regression trend
## 
## 
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.113176 -0.009113 -0.000127  0.009871  0.144195
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.304e-03 1.194e-03  1.092   0.275    
## z.lag.1     -1.068e+00 3.278e-02 -32.575  <2e-16 ***  
## tt          -1.668e-06 2.224e-06 -0.750   0.454    
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```

## Residual standard error: 0.01817 on 926 degrees of freedom
## Multiple R-squared:  0.534, Adjusted R-squared:  0.533
## F-statistic: 530.6 on 2 and 926 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -32.5753 353.7175 530.5761
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau3 -3.96 -3.41 -3.12
## phi2  6.09  4.68  4.03
## phi3  8.27  6.25  5.34

```

La p -value associée à β_1 est de $0.454 > 0.05$, on rejette H_0 , nous passons à la spécificité “drift”.

```
summary(ur.df(rtt,type="drift",lag=0))
```

```

##
## #####
## # Augmented Dickey-Fuller Test Unit Root Test #
## #####
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.113193 -0.009232 -0.000189  0.009910  0.144482
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0005282 0.0005963  0.886   0.376
## z.lag.1     -1.0673775 0.0327674 -32.574   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01817 on 927 degrees of freedom
## Multiple R-squared:  0.5337, Adjusted R-squared:  0.5332
## F-statistic: 1061 on 1 and 927 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -32.5744 530.5455
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78

```

La p -value associée à β_0 est de $0.376 > 0.05$, β_0 n'est pas significatif. On passe à la spécification “none” qui teste :

$$H_0 : \rho - 1 = 0 \quad vs \quad H_a : |\rho| < 1$$

```

summary(ur.df(rtt,type="none",lag=0))

##
## #####
## # Augmented Dickey-Fuller Test Unit Root Test #
## #####
## 
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.112652 -0.008697  0.000347  0.010428  0.145002
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1 -1.06660    0.03275 -32.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01817 on 928 degrees of freedom
## Multiple R-squared:  0.5333, Adjusted R-squared:  0.5328
## F-statistic:  1061 on 1 and 928 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -32.5661
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62

```

La valeur de t-statistique est de -37.9053 ce qui est inférieur à tau1 à 5% de -1.95. Donc on rejette H_0 et le PGD de rtt est stationnaire. Cette conclusion n'est valable que si les ϵ_t ne sont pas autocorrélés :

```
plot(ur.df(rtt,type= "none",lags=0))
```

On remarque de l'autocorrélation partielle à l'ordre 4, 6, 22 et 27. Notre conclusion n'est pas valide nous devons effectuer le test de Dickey-Fuller Augmenté.

- Dickey-Fuller Augmenté :

```

Schwert<-as.integer(12*(T/100)^(0.25))
pmax<-Schwert
pmax

## [1] 20
summary(CADFtest(rtt,criterion="BIC",type="none",max.lag.y=pmax))

## Augmented DF test
##                                     ADF test
## t-test statistic:                 -3.215494e+01
## p-value:                         4.818713e-29
## Max lag of the diff. dependent variable: 0.000000e+00

```

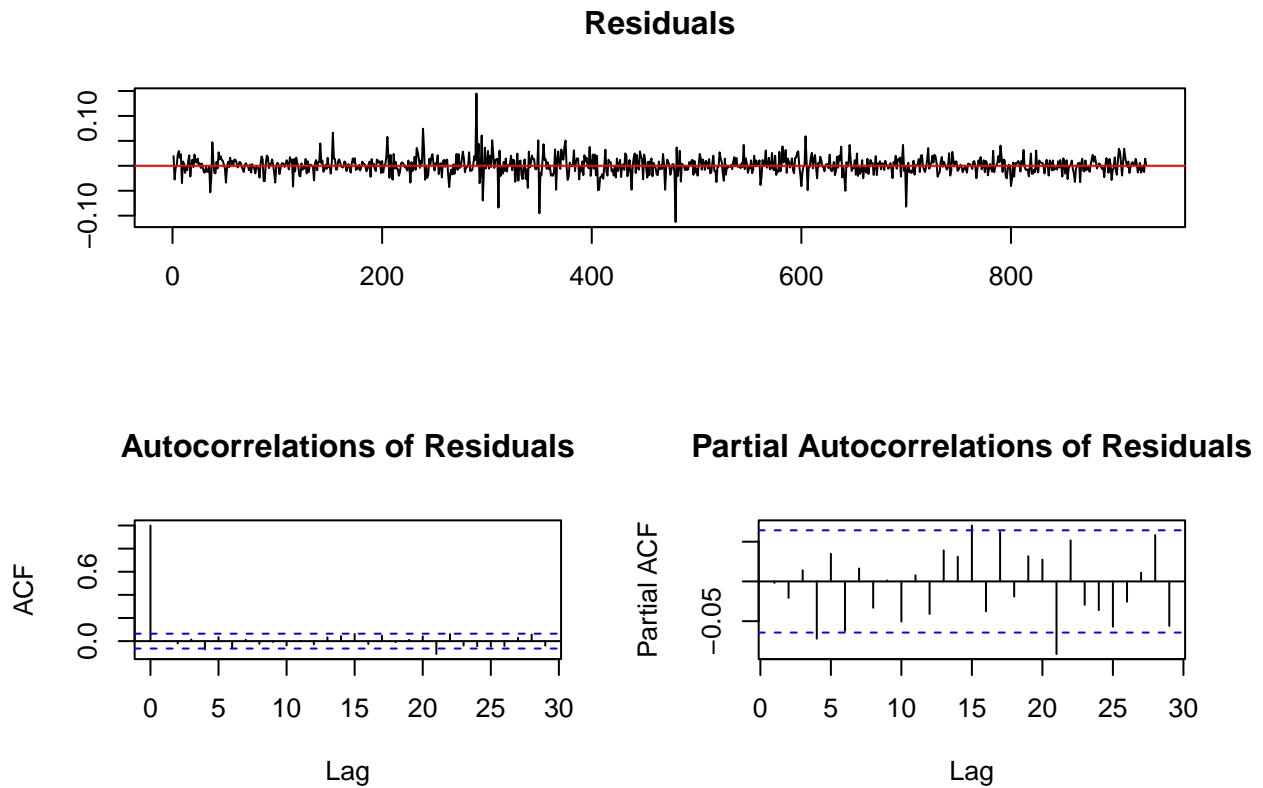


Figure 11: Corrélogramme des résidus de la régression DF *none*

```
##  
## Call:  
## dynlm(formula = formula(model), start = obs.1, end = obs.T)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -0.112625 -0.008698  0.000356  0.010328  0.144984  
##  
## Coefficients:  
##             Estimate Std. Error t value Pr(>|t|)  
## L(y, 1) -1.06484    0.03312 -32.16   <2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 0.01818 on 908 degrees of freedom  
## Multiple R-squared:  0.5324, Adjusted R-squared:  0.5319  
## F-statistic:     NA on NA and NA DF, p-value: NA
```

Nous prenons lag=0

```
summary(ur.df(rtt, type = "none", lag = 0))
```

```
##  
## #####  
## # Augmented Dickey-Fuller Test Unit Root Test #  
## #####  
##
```

```

## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1)
##
## Residuals:
##       Min        1Q     Median        3Q       Max
## -0.112652 -0.008697  0.000347  0.010428  0.145002
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## z.lag.1 -1.06660   0.03275 -32.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01817 on 928 degrees of freedom
## Multiple R-squared:  0.5333, Adjusted R-squared:  0.5328 
## F-statistic: 1061 on 1 and 928 DF, p-value: < 2.2e-16
##
##
## Value of test-statistic is: -32.5661
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62

```

La valeur de la t-statistique est de $-32.5661 < -1.95$, on rejette H_0 . Le PGD qui a générée notre série est stationnaire.

- Zivot-Andrews

```

summary(ur.za(rtt, model="both",lag=pmax))

##
## #####
## # Zivot-Andrews Unit Root Test #
## #####
##
##
## Call:
## lm(formula = testmat)
##
## Residuals:
##       Min        1Q     Median        3Q       Max
## -0.111011 -0.009485  0.000039  0.009553  0.133470
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -2.999e-03  2.367e-03 -1.267  0.20555
## y.l1         -3.501e-01  1.851e-01 -1.891  0.05890 .
## trend        3.289e-05  1.310e-05  2.510  0.01225 *  
## y.dl1        2.871e-01  1.806e-01  1.590  0.11226
## y.dl2        2.397e-01  1.758e-01  1.363  0.17324
## y.dl3        2.490e-01  1.708e-01  1.458  0.14515
## y.dl4        1.641e-01  1.657e-01  0.990  0.32223

```

```

## y.dl5      1.865e-01  1.601e-01  1.165  0.24434
## y.dl6      1.130e-01  1.548e-01  0.730  0.46568
## y.dl7      1.280e-01  1.487e-01  0.861  0.38973
## y.dl8      8.210e-02  1.421e-01  0.578  0.56355
## y.dl9      7.615e-02  1.348e-01  0.565  0.57226
## y.dl10     1.046e-02  1.277e-01  0.082  0.93473
## y.dl11     1.107e-02  1.202e-01  0.092  0.92664
## y.dl12     -4.451e-02 1.129e-01 -0.394  0.69344
## y.dl13     -1.156e-02 1.052e-01 -0.110  0.91254
## y.dl14     1.263e-02  9.746e-02  0.130  0.89695
## y.dl15     6.801e-02  8.873e-02  0.767  0.44357
## y.dl16     3.320e-02  8.032e-02  0.413  0.67947
## y.dl17     7.774e-02  7.036e-02  1.105  0.26950
## y.dl18     5.618e-02  6.043e-02  0.930  0.35281
## y.dl19     7.413e-02  4.820e-02  1.538  0.12441
## y.dl20     9.723e-02  3.309e-02  2.938  0.00338 **
## du        -8.513e-03  2.708e-03 -3.144  0.00172 **
## dt        -2.960e-05  1.365e-05 -2.169  0.03037 *
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01795 on 884 degrees of freedom
##   (21 observations deleted due to missingness)
## Multiple R-squared:  0.05472,   Adjusted R-squared:  0.02905
## F-statistic: 2.132 on 24 and 884 DF,  p-value: 0.001275
##
##
## Teststatistic: -7.2939
## Critical values: 0.01= -5.57 0.05= -5.08 0.1= -4.82
##
## Potential break point at position: 311

```

La t-statistique en valeur absolue de γ_{20} est supérieur à 1,6. La $p - value$ de δ_1 est de 0.00172 < 0.05, donc il est significatif. La $p - value$ de δ_2 est de 0.03037 < 0.05, donc il est pas significatif. La valeur de la t stat est inférieure à la valeur critique. On rejette H_0 et ρ est différent de 1. Nous avons un PGD stationnaire avec une rupture structurelle En ce qui concerne la date de rupture, nous avons “Potential break point at position: 2087”. La date de rupture est donc le 29 mars 2022 :

```
dates[1511+311]
```

```

## [1] "2022-03-29"
• Lee-Strazicich
LS_test <- ur.ls(y=rtt, model = "crash", breaks = 1, lags = 1, method="GTOS", pn = 0.1, print.results =
##
## [1] -32.3034
## [1] "First possible structural break at position: 380"
## [1] "The location of the first break - lambda_1: 0.4 , with the number of total observations: 930"
## Critical values - Crash model:
##       1%      5%      10%
## [1,] -4.239 -3.566 -3.211
## [1] "Number of lags determined by general-to-specific lag selection: NA"
## Runtime:
## Time difference of 0.02167636 mins

```

Il faut introduire 1 variable supplémentaire pour prendre en compte l'autocorrélation dans rtt .

9) Conclusion

On a montré que :

- notre série rtt est asymétrique (propriété 1)
- la distribution de rtt est leptokurtique (propriété 2)
- La série rtt^2 présente de fortes auto-corrélations et notre série rtt présente de l'autocorrélation faible (propriété 3)
- notre série rtt présente des clusters de volatilité (propriété 4)
- Les queues de distribution des aléas de notre ARMA-GARCH sont plus épaisses que celles d'une loi normale (propriété 5)
- notre série rtt ne présente pas d'effet de levier (propriété 6)
- notre série rtt présente un effet week-end (propriété 7)
- le PGD de notre série rtt est stationnaire (propriété 8)

10) Tableau récapitulatif

Propriétés	RTE	RTT
Propriété 1	✓	✓
Propriété 2	✓	✓
Propriété 3	✓	✓
Propriété 4	✓	✓
Propriété 5	✓	✓
Propriété 6	✓	✗
Propriété 7	✓	✓
Propriété 8	✓	✓

Table 1: Tableau des propriétés de *rte* et *rtt*