COUSTILLAC Cêlestine

Etude des 8 caractéristiques statistiques du rendement logarithmique l'action Korea Aerospace Industries

Introduction

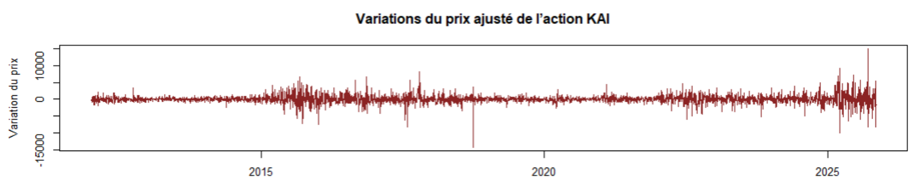
Dans le cadre de ce projet, nous étudions les **8 caractéristiques statistiques du rendement logarithmique d'une action cotée distinguées par Charpentier** (2002). L'objectif est d'analyser ces propriétés sur une première période, puis de vérifier si elles se retrouvent dans une période plus récente. Cette comparaison entre les deux périodes est indispensable, car par la suite nous devrons estimer la Value at Risk (VaR) et vérifier, à l'aide du backtesting, si cette estimation reste cohérente avec les données observées. Pour que ces méthodes soient fiables, il est important que les propriétés statistiques identifiées sur la période d'estimation se retrouvent également dans la période de test.

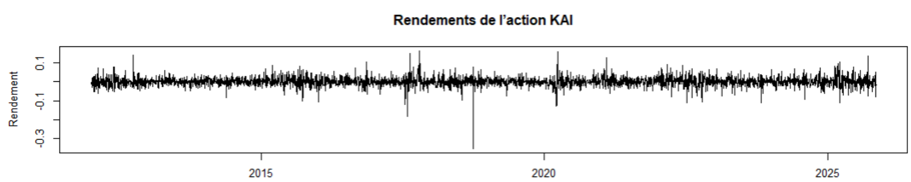
L'action retenue pour cette étude est **Korea Aerospace Industries** (047810.KS), un acteur majeur de l'industrie aéronautique sud-coréenne. L'entreprise conçoit et assemble des avions militaires, hélicoptères, drones et structures aéronautiques pour divers programmes nationaux et internationaux. Sa présence dans un secteur fortement technologique, sensible aux contrats publics, aux cycles géopolitiques et aux annonces industrielles crée une dynamique de marché riche et volatile, ce qui en fait un support pertinent pour une analyse statistique de rendements financiers.

Pour mener l'étude, les données quotidiennes de cours seront séparées en deux périodes distinctes :

1. Une **période d'estimation (r_{te})** : du 1er janvier 2012 au 30 décembre 2020
2. Une **période de test (r_{te})** : du 4 janvier 2021 au 7 novembre 2025

Ce découpage permet d'observer si les propriétés mises en évidence au cours de la période d'estimation se retrouvent dans un contexte plus récent, et donc de préparer efficacement les travaux d'estimation et de validation de la VaR.





L'observation des trois graphiques montre que le cours ajusté de l'action KAI présente une **tendance globale croissante**, malgré une phase baissière entre 2016 et 2020.

Les **variations de prix et les rendements restent centrés autour de zéro** sans tendance particulière. De plus, les **fluctuations autour de 0 changent clairement au cours du temps**, particulièrement pour dpt : certaines périodes sont calmes tandis que d'autres sont beaucoup plus agitées. Il semble y avoir des paquets de volatilité en 2016 et en 2025.

Règle de décision :

Dans tout le projet, nous utiliserons la règle de décision suivante au seuil de risque $\alpha = 5\%$:

- *Soit on regarde la p-value : si $p - value \ < \ 0{,}05$ alors on rejette H_0*
- *Soit on regarde la statistique de test t : si $|t| < 1{,}96$ alors on rejette H_0*

Analyse des 8 propriétés statistiques du rendement logarithmique de la série r_{te}

Dans cette partie, on étudie les principales propriétés statistiques du rendement logarithmique r_{te} sur l'ensemble d'estimation. *(Annexe p.12)*

On commence par calculer la **moyenne empirique**, on trouve -0.0001622717.

Est-ce que cette moyenne empirique proche de 0 est statistiquement nulle ? Autrement dit, Est-ce que l'espérance du processus ayant généré r_{te}, notée μ, est nulle ?

Pour cela, nous allons tester : $H_0 \colon E(r_{te}) = \ \mu = \ 0$ versus $H_a \colon E(r_{te}) = \ \mu \neq 0$

On obtient une $p - value \ = \ 0.758 \ > \ 0.05$, alors on accepte H_0. Ainsi, **l'espérance du PGD qui a généré r_{te} est nulle.**

Propriété 1 : Asymétrie perte/gain

Afin d'analyser l'asymétrie de la distribution des rendements, nous commençons par tester la symétrie de la loi de r_{te} à l'aide d'un **test sur le skewness**. Plus précisément, il s'agit de vérifier si le troisième moment centré normalisé (skewness) est nul, ce qui correspond à une distribution symétrique, ou s'il est significativement différent de zéro, ce qui traduirait une asymétrie des rendements. *(Annexe p.12)*

Pour cela, nous allons tester : $H_0 \ : \ E\left[\left(\frac{X-E(X)}{\sigma_X}\right)^3\right] = 0$ versus $H_a \ : \ E\left[\left(\frac{X-E(X)}{\sigma_X}\right)^3\right] \neq 0$

Les résultats du test d'Agostino indiquent une $p - value \ < \ 2.2e^{-16}$. La p-value étant largement inférieure au seuil de 5 %, la skewness est donc statistiquement différente de zéro, ce qui signifie que la **distribution des rendements n'est pas symétrique**.

Ainsi, nous pouvons interpréter sa valeur calculée : $skew \ = \ -1.2156 \ < \ 0$, cela traduit une **asymétrie orientée vers la gauche** : on observe généralement de nombreux rendements positifs mais de faibles valeurs, tandis que les rendements négatifs, plus rares, sont de fortes valeurs. Autrement dit, les petites hausses sont fréquentes mais les baisses, lorsqu'elles surviennent, sont plus importantes, ce qui caractérise une distribution avec une queue gauche plus lourde.

Propriété 2 : Queues de distribution épaisses

Pour la seconde propriété, nous nous intéressons à la **forme des queues de la distribution** à travers l'analyse de la **kurtosis**. L'objectif est de déterminer si la distribution des rendements présente des queues plus épaisses ou plus fines que celle d'une loi normale, en testant si son quatrième moment centré normalisé s'écarte de la valeur théorique 3. *(Annexe p.13)*

Pour cela, nous allons tester : $H_0 : E\left[\left(\frac{X-E(X)}{\sigma_X}\right)^4\right] = 0$ versus $H_a : E\left[\left(\frac{X-E(X)}{\sigma_X}\right)^4\right] \neq 3$

Le **test d'Anscombe-Glynn** renvoie une $p-value < 2.2e^{-16}$. La p-value étant largement inférieure au seuil de 5 %, nous rejetons l'hypothèse nulle selon laquelle la kurtosis serait égale à 3. La distribution de r_{te} présente donc une kurtosis significativement différente de celle d'une loi normale.

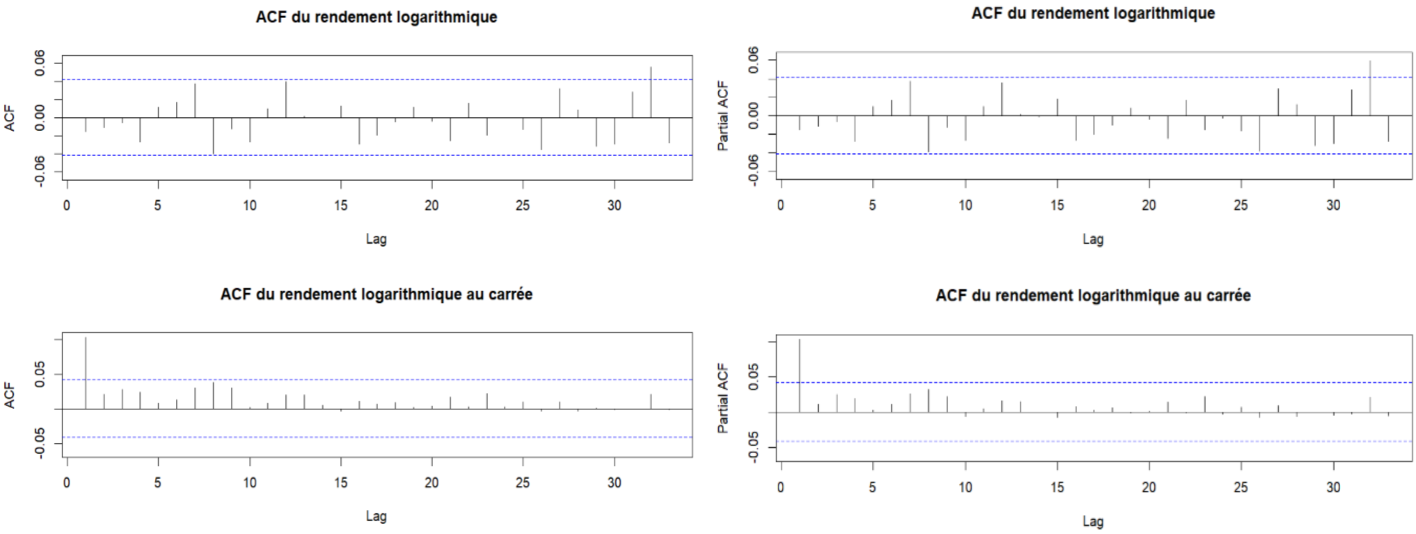
Ainsi, nous pouvons interpréter sa valeur calculée : $kurt = 27.073 \gg 3$, cela indique une distribution fortement **leptokurtique**, c'est-à-dire caractérisée par des queues beaucoup plus épaisses que celles d'une loi normale. *(voir sortie code p.12 annexe)*

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

Afin d'examiner la troisième propriété, nous analysons maintenant la **présence éventuelle d'autocorrélation dans les rendements et dans les rendements au carré**.

3.1 – ACF & PACF

Les graphiques ci-dessous représentent les coefficients d'autocorrélation totaux ρ_k des rendements et des rendements au carré pour différents retards :



L'analyse des ACF et PACF montre que, pour les rendements logarithmiques, seul ρ_{32} dé passe légèrement l'intervalle de confiance, ce qui indique une autocorrélation significative mais faible en raison de sa petite valeur (< 0.06). En revanche, pour les rendements au carré, c'est uniquement l'autocorrélation d'ordre 1, ρ_1 ($\cong$ 0.1), qui dépasse de l'intervalle de confiance, ce qui met en évidence une autocorrélation de court terme plus marquée mais toujours modérée.

3.2 - Test de Ljung-Box

Pour confirmer ces observations, nous complétons l'analyse avec la statistique de **Ljung-Box**, qui teste l'hypothèse d'absence d'autocorrélation jusqu'à un ordre donné. Ce test global permet de déterminer s'il existe au moins un retard pour lequel la corrélation est significative. *(Annexe p.14)*

Pour cela, nous allons tester :

$$H_0 : \rho(k) = 0 \text{ pour } k = 1, \dots, K \quad \text{versus} \quad H_a : \rho(k) \neq 0 \text{ pour au moins une valeur de } k \in \{1, \dots, K\}$$

Pour les rendements au carré, le test de Ljung–Box au lag 1 fournit une p-value égale à $1.134e^{-06}$, nettement inférieure au seuil de 5 %. Nous rejetons donc H_0 et concluons à la présence d'autocorrélation dès le premier retard dans la série des rendements au carré.

Pour les rendements, l'examen des p-values successives du test de Ljung–Box jusqu'au lag 40 montre que toutes les p-values sont supérieures à 5 %, à l'exception de celle associée au lag 40, égale à $0{,}0138 < 5\%$. On rejette donc H_0 au lag 40, ce qui met en évidence l'existence d'au moins une autocorrélation significative dans r_{te}. (*voir sortie de code p.14)*

Pour modéliser cette caractéristique nous utiliserons un modèle ARMA(p,q).

3.3 - EACF

Dans un premier temps, nous devons déterminer la valeur de p et q du ARMA(p,q) via l'eacf.

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

On trouve p = 2 et q = 2

3.4 - Arima

Dans un second temps, nous faisons l'estimation du modèle ARMA(p,q) avec les valeurs trouvées via l'eacf. L'eacf suggère initialement un modèle ARMA(2,2), mais ce modèle ne converge pas numériquement. Nous testons donc un modèle ARMA(1,1), fréquemment pertinent dans les séries financières, mais lui aussi ne converge pas. *(Annexe p.15 et p.27)*

Comme l'ACF des rendements logarithmiques présentait une autocorrélation marquée à l'ordre 32, nous testons un modèle AR(23) et MA(23). *(Annexe p.16 à p.27)*

Finalement le **MA(23)** sera notre unique modèle acceptable.

- Les coefficients du MA(23) sont tous significatifs :

```
z test of coefficients:

     Estimate Std. Error z value Pr(>|z|)
ma12 0.043073   0.021034  2.0477 0.040587 *
ma32 0.056104   0.020735  2.7058 0.006814 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

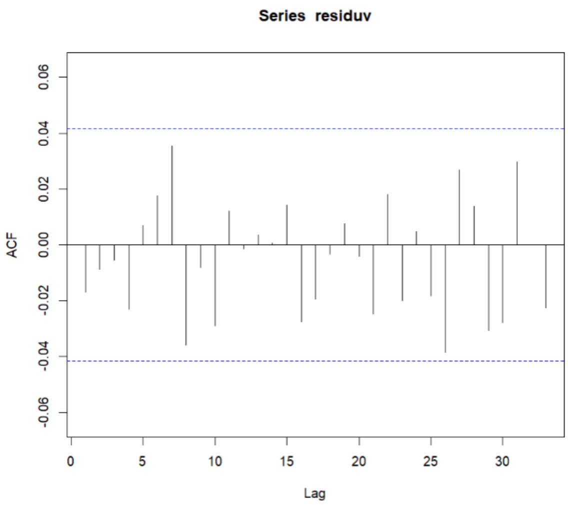
- De plus, les tests menés sur les résidus confirment la validité du modèle :
  1. Le test de Student ne rejette pas l'hypothèse d'espérance nulle des aléas :

$$H_0 : E[\varepsilon] = 0 \quad \text{versus} \quad H_a : E[\varepsilon] \neq 0$$

```
##
##  One Sample t-test
##
## data:  residu
## t = -0.27562, df = 2212, p-value = 0.7829
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0011750579  0.0008854589
## sample estimates:
##     mean of x
## -0.0001447995
```

Comme on a une $p - value \; = \; 0.7829 \; > \; 0.05$ donc on accepte H0

  2. Les tests de Ljung–Box, ainsi que le corrélogramme des résidus standardisés, ne mettent pas en évidence d'autocorrélation résiduelle : on n'a aucun ρ qui sort et toutes les p-values de Ljung–Box sont supérieur à 5%.




```
 [1] 0.4278902 0.6707695 0.8334948 0.7268137
 [5] 0.8277271 0.8294639 0.5864363 0.3891602
 [9] 0.4735125 0.3986888 0.4590287 0.5445619
[13] 0.6235977 0.6978828 0.7304352 0.6729933
[17] 0.6781011 0.7375565 0.7839239 0.8288573
[21] 0.8031209 0.8099780 0.8087731 0.8461210
[25] 0.8500402 0.7356629 0.6993180 0.7252569
[29] 0.6651915 0.6237546 0.5705070 0.6204045
[33] 0.6114146 0.6545330 0.6449551 0.5498013
[37] 0.4611742 0.4754691 0.3825357 0.1609895
```

→ Nous retenons donc le modèle MA(32).

Propriété 4 : Clusters de volatilité

Nous analysons maintenant la quatrième propriété, qui concerne l'existence de **clusters de volatilité**. Ce phénomène, remet en cause l'hypothèse d'homoscédasticité conditionnelle. Afin de vérifier formellement si la variance dépend des chocs passés, nous appliquons le **test ARCH d'Engle (1982).** *(Annexe p.28)*

Pour cela, nous allons tester :

$$H_0 : homoscédasticité \;\; conditionnelle \quad \text{VS} \quad H_a : hétéroscédasticité \; conditionnelle$$

On rejette l'hypothèse nulle d'homoscédasticité conditionnelle à l'ordre 1 ($p - value \ = \ 1.159e - 06 \ >$ 5%). Cela signifie que la variance conditionnelle des rendements n'est pas constante : elle dépend des chocs passés, ce qui confirme la présence d'**hétéroscédasticité conditionnelle** et donc de **clusters de volatilité** déjà observés visuellement sur le chronogramme. (voir sortie code p.26 annexe)

Propriété 5 : Queues épaisses conditionnelles

Après avoir estimé l'équation de la moyenne avec un modèle MA(32) et vérifié que ses aléas ne présentent plus d'autocorrélation, le test d'Engle met en évidence la présence de volatilité conditionnelle. Il est donc nécessaire de modéliser la variance avec un modèle GARCH, en particulier le GARCH(1,1).

Dans un premier temps, nous avons estimé un modèle GARCH(1,1) sur les résidus standardisés à l'aide de la fonction garch, mais l'algorithme ne convergeait pas. Nous avons donc réestimé le même modèle avec la fonction garchFit, toujours en GARCH(1,1). *(Annexe p.28 à p.31)*

```
Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)    
omega    0.06045     0.01156    5.229  1.7e-07 ***
alpha1   0.14304     0.01690    8.463  < 2e-16 ***
beta1    0.80983     0.02055   39.410  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Les coefficients du modèle GARCH(1,1) étant significatifs (p-value < 5%), nous vérifions maintenant si ce modèle a **bien capturé toute l'hétéroscédasticité conditionnelle** présente dans la série.

```
	ARCH LM-test; Null hypothesis: no ARCH effects

data:  resvolat
Chi-squared = 3.4542, df = 1, p-value = 0.06309

	ARCH LM-test; Null hypothesis: no ARCH effects

data:  resvolat
Chi-squared = 11.081, df = 20, p-value = 0.9441
```

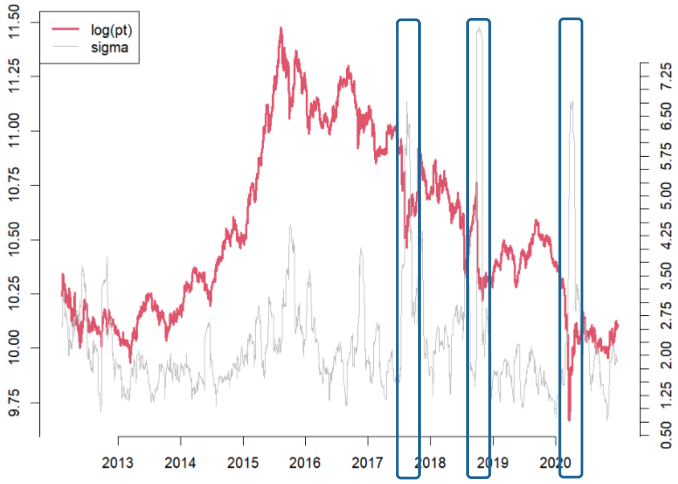
Les tests ARCH appliqués aux résidus standardisés du modèle GARCH(1,1) fournissent des p-values supérieures à 5 %, ce qui conduit à accepter l'hypothèse d'**absence d'effets ARCH résiduels**. On en déduit que le modèle GARCH(1,1) parvient à capturer correctement la majeure partie de l'hétéroscédasticité conditionnelle de la série.

Enfin, le test d'Anscombe–Glynn conduit à rejeter l'hypothèse de kurtosis égale à 3 (p-value<2.2e-16 <0.05). Ainsi, nous pouvons interpréter la kurtosis estimée ($kurt \ = \ 11.657$) est largement supérieure à 3, ce qui traduit une **distribution fortement leptokurtique**. Ainsi, même après la prise en compte de la volatilité conditionnelle via le modèle GARCH(1,1), les résidus conservent des queues épaisses, confirmant la présence de leptokurtose conditionnelle.

Propriété 6 : Effet de levier

Dans cette partie, nous cherchons à mettre en évidence **l'effet de levier**, c'est-à-dire l'asymétrie entre l'impact des baisses et des hausses du cours sur la volatilité. Pour cela, comme estimateur naïf de la volatilité à la date (t), on utilise l'écart-type des 22 jours les plus récents, c'est-à-dire un **sigma récursif** calculé sur une fenêtre glissante de 22 rendements.

On semble apercevoir sur le graphique 3 clusters de volatilités.



De plus, à partir du tableau ci-dessous on observe que sigma est **presque deux fois plus élevée** lors de la plus forte baisse que lors de la plus forte hausse. Ça illustre bien l'**effet de levier** : les **mauvaises nouvelles** s'accompagnent d'une **forte hausse de volatilité**, alors que les bonnes nouvelles (grosse hausse du cours) font monter la volatilité, mais de façon beaucoup plus modérée.

Type	Date	Rendement	Sigma
Plus forte baisse	2018-09-28	-14110.027	8.050139
Plus forte hausse	2017-10-19	8049.332	4.330013

La date du 28 septembre 2018 correspond à l'annonce de la perte du contrat T-X de l'US Air Force, qui a provoqué un effondrement du cours de KAI, alors que celle du 9 octobre 2017 coïncide avec une phase de reprise de confiance lors du salon ADEX 2017.

Propriété 7 : La saisonnalité

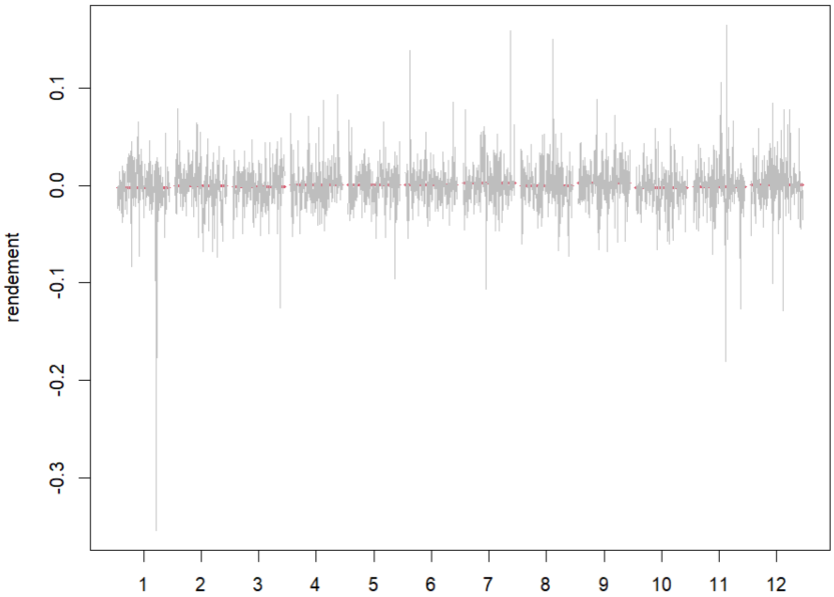
7.1 - Journalier - Effet week-end

Dans la littérature, plusieurs travaux montrent que l'accumulation d'informations pendant les périodes de fermeture augmente la volatilité, en particulier en début de semaine (French & Roll, 1986) ou à partir du mercredi (Baillie & Bollerslev, 1989).

	lundi	mardi	mercredi	jeudi	vendredi
moyenne en %	-0.1372707	-0.04860218	0.2792566	-0.04907711	-0.1255857
écart-type annuel en %	37.0181889	34.09480116	39.4719837	37.58796588	47.0498409
skewness	0.1263369	-0.05324389	1.8511301	-1.08039631	-4.0950068
kurtosis	4.5372373	1.75826526	10.2467495	9.71205625	46.1047988

La **moyenne de mercredi** se distingue par son signe positif contrairement aux autres. Les **volatilités**, elle, deviennent plus élevées à partir du mercredi et atteignent même un maximum **le vendredi**, ce qui correspond davantage aux observations de **Baillie & Bollerslev (1989).** On observe donc bien un effet week-end.

7.2 - Mensuelle - Effet janvier



Nous avons la **volatilité des rendements** en **janvier** qui est beaucoup plus importante que les autres mois.

Concernant les rendements moyens, les plus élevés apparaissent en juillet et en septembre, tandis que les mois les plus faibles paraissent être en janvier et décembre. Cependant, **aucune moyenne ne se distingue réellement**.

Propriété 8 : Stationnarité

Pour étudier la stationnarité du processus générateur des rendements logarithmiques r_{te}, nous appliquons une série de tests de racine unitaire. Nous commençons par le test de Dickey–Fuller classique, puis nous vérifions la validité de ses hypothèses à l'aide du Dickey–Fuller augmenté (ADF). Enfin, pour tenir compte de la possibilité d'une rupture structurelle au cours de la période étudiée, nous complétons l'analyse avec les tests de Zivot–Andrews et de Lee–Strazicich. *(Annexe p.31 à p.41)*

8.1 - Test de Dickey-Fuller (DF)

Nous commençons par appliquer le test de Dickey–Fuller en partant de la spécification « **trend** ».

$$H_0 : \rho - 1 = 0 \;\&\; \beta_1 = 0 \quad \text{versus} \quad H_a : |\rho| < 1 \;\&\; \beta_1 \neq 0$$

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.445e-04  1.054e-03   0.422    0.673
z.lag.1     -1.016e+00  2.128e-02 -47.749   <2e-16 ***
tt          -5.504e-07  8.254e-07  -0.667    0.505
```

Dans cette spécification « trend », β_1 n'est pas significatif ($p - value = 0.505 > 5\,\%$), ce qui conduit à supprimer la tendance et à considérer ensuite la spécification « **drift** ».

$$H_0 : \rho - 1 = 0 \;\&\; \beta_0 = 0 \quad \text{versus} \quad H_a : |\rho| < 1 \;\&\; \beta_0 \neq 0$$

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0001645  0.0005270  -0.312    0.755
z.lag.1     -1.0158153  0.0212735 -47.750   <2e-16 ***
```

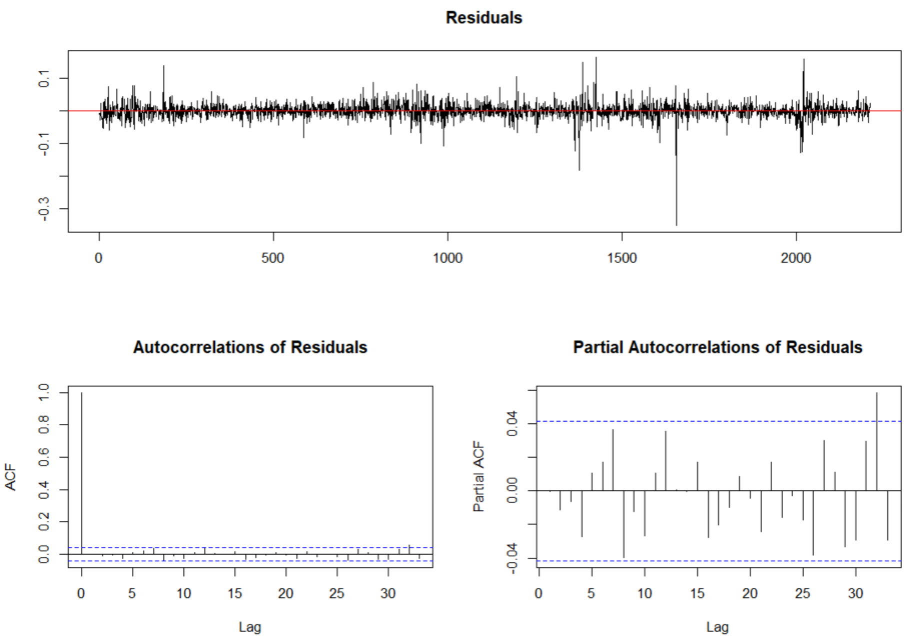
Dans cette seconde spécification, la constante (β_0) n'est elle non plus pas significative, ce qui conduit finalement à considérer la spécification « **none** ».

$$H_0 : \rho - 1 = 0 \; versus \; H_a : \rho - 1 \neq 0$$

```
        Estimate Std. Error t value Pr(>|t|)
z.lag.1 -1.01577    0.02127  -47.76   <2e-16 ***
```

Nous pouvons donc effectuer le test de racine unitaire. Comme la statistique de test (-47.7589) est très inférieure à la borne critique (-1.95), alors on rejete l'hypothèse de racine unitaire et donc à la **stationnarité du PGD**.

Cependant, cette conclusion n'est valide que si les aléas de la régression de Dickey et Fuller ne sont pas auto-corrélés.



La figure nous indique que **les aléas sont auto-corrélés aux ordres 32**. C'est l'autocorrélation à l'ordre 32 déjà observé avec l'ACF de r_{te} qui réapparaît dans les résidus : le modèle de Dickey–Fuller ne parvient pas à la capturer et cette autocorrélation se retrouve donc rejetée dans le terme d'erreur.

Ainsi, notre **conclusion** concernant l'absence de racine unitaire (RU) n'est **pas valide**. Nous devons effectuer un test de RU dans le cadre de la régression de Dickey Fuller Augmenté (ADF).

8.2 - Test de Dickey Fuller Augmenté (ADF)

Pour le test de **Dickey–Fuller augmenté**, nous commençons par déterminer un nombre maximal de retards à partir de la règle de Schwert, ce qui conduit ici à un maximum de 26 retards possibles.

Dans un premier temps, nous souhaitons vérifier que la constante n'est pas significative dans la spécification « drift ». Le critère MAIC, suggère un lag optimal de 6. Nous vérifions alors la significativité de la constante dans ce modèle ADF avec dérive (« drift ») et 6 retards : le coefficient B_0 présente une p-value de $0{,}772\ >\ 5$ %, ce qui confirme que la spécification appropriée est bien "none".

Nous passons donc à la spécification none. Le critère MAIC suggère de nouveau un lag optimal de 6. Dans ce modèle ADF avec 6 retards, la spécification est validée en s'assurant que le dernier coefficient de retard γ_6 est significatif : sa statistique t | $-1{,}771$ | est supérieure au seuil 1,64 (au niveau 10 %), ce qui justifie la présence de ce retard dans l'équation.

On peut alors procéder au test de racine unitaire. La statistique de test vaut $t = -17{,}0962$, valeur très inférieure à la borne critique à 5 % (environ –1,95). Nous rejetons donc l'hypothèse de racine unitaire et concluons, à partir du test ADF, que le **PGD est stationnaire**.

8.3 - Test de Zivot et Andrews (ZA)

Cependant, ces résultats ne sont valides que si pendant toute la période d'étude, il n'y a pas de choc structurelle ou conjoncturelle. Or, en présence d'un changement de régime, ces tests peuvent conduire à des conclusions biaisées. Pour en tenir compte, nous utilisons le test de Zivot–Andrews.

Pour cela, nous allons tester : H_0 : *DS sans date de rupture* VS H_a : *TS avec une date de rupture*

Nous retenons tout d'abord le modèle le plus général, dit **« both »**, avec un nombre de retards initial égal à Schwert (26). Dans cette spécification, le dernier coefficient de retard n'est pas significatif (statistique $t = 1{,}032$ en valeur absolue < 1,64), ce qui conduit à réduire progressivement le nombre de retards.

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.445e-03  1.789e-03  -1.925 0.054303 .
y.l1        -3.479e-01  1.227e-01  -2.835 0.004629 **
trend        1.139e-05  3.518e-06   3.239 0.001220 **
y.dl1        3.256e-01  1.197e-01   2.719 0.006598 **
y.dl2        3.025e-01  1.168e-01   2.591 0.009634 **
y.dl3        2.918e-01  1.137e-01   2.566 0.010355 *
...

y.dl25       4.594e-02  2.143e-02   2.144 0.032157 *
du          -8.729e-03  2.271e-03  -3.844 0.000124 ***
dt          -1.035e-05  3.937e-06  -2.628 0.008653 **
```

Dans cette spécification, les coefficients δ_1 $(p - value = 0.0007 < 0.05)$, δ_2 $(p - value = 0{,}0086 < 0.05)$ et β_1 $(p - value = 0.0012 < 0.05)$ sont significatifs.

Nous avons donc la bonne spécification, nous pouvons passer au test de racine unitaire. La statistique de test de racine unitaire ($t = -10.9818$) est alors très inférieure à la borne critique (-5.08), ce qui amène à rejeter l'hypothèse de racine unitaire : le PGD est TS avec une date de rupture.

Cependant, comme le rendement n'a pas de tendance alors il ne peut pas être TS, le **PGD est donc stationnaire avec une date de rupture autour du 10 août 2015.**

Cette rupture coïncide avec plusieurs décisions importantes impliquant l'État coréen et KAI. D'une part, au printemps–été 2015, le gouvernement signe avec KAI les contrats de développement des hélicoptères légers LAH/LCH, pour un montant d'environ 1,6 billion de wons, ce qui soutient fortement le titre. D'autre part, le 5 août 2015, la DAPA notifie à KAI une restriction de trois mois à la participation aux appels d'offres publics (du 11 août au 10 novembre) à la suite d'irrégularités dans certains marchés de défense, annonce faite alors même que l'action se rapproche de 100 000 wons. Ces décisions publiques, à la fois porteuses et pénalisantes pour l'entreprise, offrent ainsi une justification économique plausible à la rupture de structure détectée autour du 10 août 2015.

8.4 - Test de Lee et Strazicich (LS)

Cependant, nous pouvons rejeter/accepter H_0 dans ZA alors qu'il y a la possibilité d'un PGD DS avec une date de rupture.

Pour compléter l'analyse de Zivot–Andrews, nous appliquons donc le test de Lee et Strazicich (LS).

Pour cela, nous allons tester : H_0 : *DS avec une date de rupture* VS H_a : *TS avec une date de rupture*

Avec un nombre de retards fixé par la règle de Schwert, la statistique de test obtenue ($t = -47.90875$) est inférieure à la valeur critique (-4.50), ce qui conduit à rejeter l'hypothèse nulle de racine unitaire et à conclure, dans un premier temps, à un PGD TS avec une date de rupture.

Cependant, comme on a absence de tendance dans la série de rendements alors il ne peut pas être TS, le **PGD est donc stationnaire avec une date de rupture avec une date de rupture autour du 30 décembre 2020.** Cette date coïncide avec la signature d'un contrat majeur de production de KUH-1 Surion avec l'agence d'armement DAPA (environ 1 000 milliards de wons).

Comparaison

Pour synthétiser l'analyse, nous comparons maintenant les huit propriétés stylisées du rendement logarithmique sur la période d'estimation r_{te} et sur la période de test r_{tt}, afin de voir dans quelle mesure le comportement de l'action KAI est resté stable ou s'est modifié au cours du temps.

(Annexe p.42 à p.81 pour étude de r_{tt})

Propriété	r_{te}	r_{tt}
1. Asymétrie perte/gain	Oui, asymétrie marquée vers la gauche	Non, absence d'asymétrie
2.Queues de distribution épaisses	Oui, queues très épaisses, kurtosis environ de 27 $\gg$ 3 (leptokurtose très forte)	Oui, queues épaisses, kurtosis environ de 5.8 $\gg$ 3 (leptokurtose modérée)
3. Autocorrélations des carrés des rendements forts et faibles pour les rendements	Oui, autocorrélations des carrés des rendements et des rendements	Oui, autocorrélations des carrés des rendements et des rendements
4. Clusters de volatilité	Oui, présence de clusters de volatilité	Oui, présence de clusters de volatilité
5. Queues épaisses conditionnelles	Oui, GARCH(1,1) capte l'hétéroscédasticité mais les résidus restent fortement leptokurtiques	Oui, GARCH(1,1) capte l'hétéroscédasticité mais les résidus restent leptokurtiques
6. Effet de levier	Oui, effet de levier	Oui, effet de levier
7. La saisonnalité	Oui, Journalier : moyenne la plus élevée le mercredi ; volatilité la plus forte le vendredi. Mensuel : volatilité la plus élevée en janvier et décembre.	Oui, Journalier : moyenne et volatilité les plus élevées le lundi. Mensuel : moyenne la plus élevée en janvier, volatilité aussi renforcée en début d'année.
8. Stationnarité	Oui, stationnaire, avec date de rupture	Oui, stationnaire, avec date de rupture

Annexe

I - Analyse des 8 propriétés statistiques du rendement logarithmique de la série rte

```r
s=sd(rte) # écart type estimé
rbar<-mean(rte) # moyenne empirique
rbar
```

```
## [1] -0.0001622717
```

```r
# Est-ce que cette moyenne empirique proche de 0 est statistiquement nulle ?
# H0 : E[rte] = mu = 0 VS Ha : E[rte] = mu =/= 0
t.test(rte)
```

```
##
##  One Sample t-test
##
## data:  rte
## t = -0.30812, df = 2212, p-value = 0.758
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0011950553  0.0008705119
## sample estimates:
##     mean of x
## -0.0001622717
```

```r
# Comme p-value = 0.758 > 0.05, alors on accepte H0.
# Ainsi, l'espérance du PGD qui a généré  rte est nulle.
```

PROPRIETE 1 - Asymétrie perte/gain

```r
library(moments)
```

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

```
##
##  D'Agostino skewness test
##
## data:  rte
## skew = -1.2156, z = -18.7279, p-value < 2.2e-16
## alternative hypothesis: data have a skewness
```

```r
# p-value < 2.2e-16 < 5 %, la skewness est donc statistiquement différente de zéro,
# ce qui signifie que la distribution des rendements n'est pas symétrique.

# Ainsi, nous pouvons interpréter sa valeur calculé : skew = -1.2156 < 0, cela
# traduit une asymétrie orientée vers la gauche : on observe généralement de
# nombreux rendements positifs mais de faible valeurs, tandis que les rendements
# négatifs, plus rares, sont de forte valeurs. Autrement dit, les petites hausses
# sont fréquentes mais les baisses, lorsqu'elles surviennent, sont plus importantes,
# ce qui caractérise une distribution avec une queue gauche plus lourde.
```

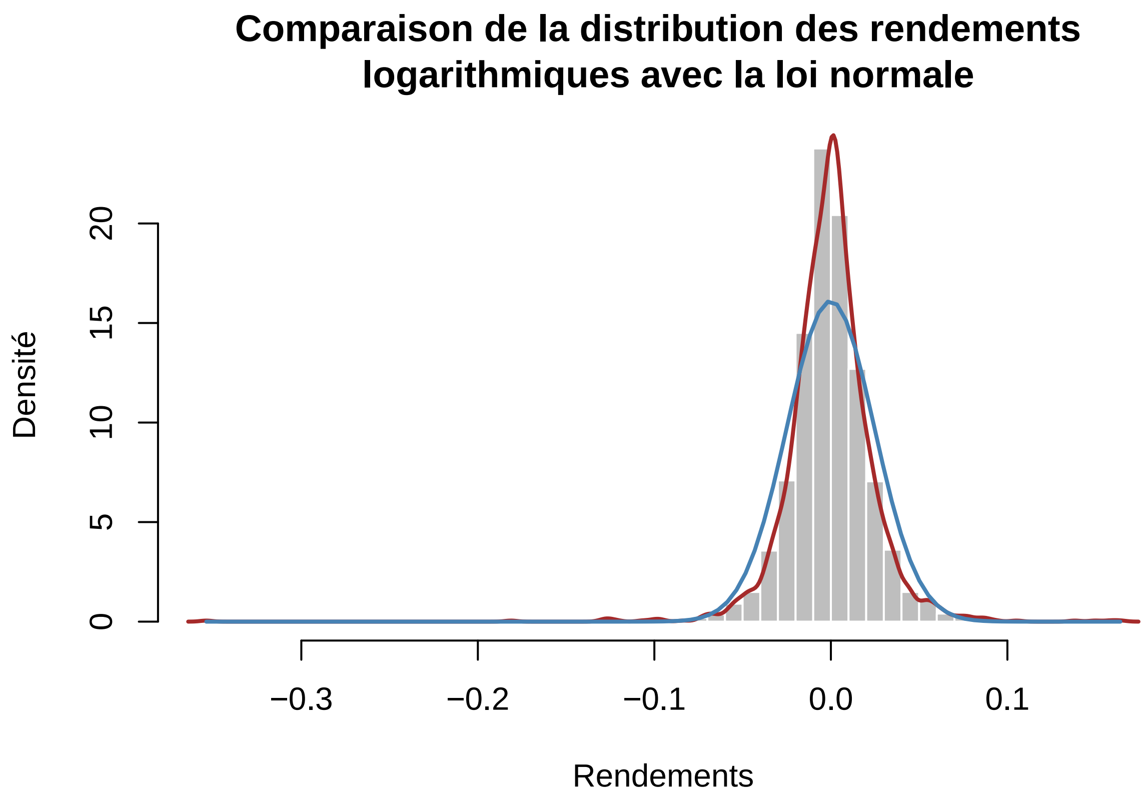
PROPRIETE 2 : Queues de distribution épaisses

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

```
##
##  Anscombe-Glynn kurtosis test
##
## data:  rte
## kurt = 27.073, z = 24.438, p-value < 2.2e-16
## alternative hypothesis: kurtosis is not equal to 3
```

```r
# p-value < 2.2e-16 < 5 %, nous rejetons l'hypothèse nulle selon laquelle la kurtosis
# serait égale à 3.

# Ainsi, nous pouvons interpréter sa valeur calculé : kurt = 27.073 >> 3, cela
# indique une distribution fortement leptokurtique, i.e. une distribution dont
# la cloche est plus pointue que celle de la loi gaussienne, avec des queues
# plus importante.
```

```r
# Comparaison de rte avec une loi normale :
mu <- mean(rte, na.rm = TRUE)
sigma <- sd(rte, na.rm = TRUE)

hist(rte, breaks = 70, freq = FALSE, col = "grey",
     main = "Comparaison de la distribution des rendements \n logarithmiques avec la loi normale",
     xlab = "Rendements",
     ylab = "Densité",
     border = "white")

lines(density(rte), col = "brown", lwd = 2)

curve(dnorm(x, mean = mu, sd = sigma),
      from = min(rte), to = max(rte),
      add = TRUE, col = "steelblue", lwd = 2)
```




```
# On observe bien la distribution leptokurtique
```

PROPRIETE 3 : Autocorrélations des carrés des rendements fortes et faibles pour les rendements

```
library(FinTS)
library(TSA)
library(lmtest)
library(forecast)
```

Analyse de la présence éventuelle d'autocorrélation dans les rendements et dans les rendements au carré

Test de Ljung-Box

```
# Rendement logarithmique au carrée
Box.test(rte^2,lag=1,type="Ljung-Box")
```

```
##
```

```
##  Box-Ljung test
##
## data:  rte^2
## X-squared = 23.686, df = 1, p-value = 1.134e-06
```

```
# p-value = 1.134e-06 < 5%
# On rejette donc l'hypothèse nulle d'absence d'autocorrélation
```

```
# Rendement logarithmique
pvaluesrt =rep(0,40)
pvaluesrt <- numeric(40)
for (i in 1:40) {
  pvaluesrt[i] = Box.test(rte,lag=i,type="Ljung-Box")$p.value
}
pvaluesrt
```

```
##  [1] 0.45677140 0.66039070 0.82418474 0.63827068 0.73069557 0.75306480
##  [7] 0.48317123 0.26119331 0.31794836 0.28154861 0.34488407 0.20522808
## [13] 0.26533016 0.33158868 0.37640713 0.32353181 0.33628109 0.39710696
## [19] 0.44491607 0.50698941 0.48019474 0.50744994 0.51809963 0.57713605
## [25] 0.60973814 0.50191806 0.43452004 0.48011182 0.41538516 0.37091796
## [31] 0.33635691 0.13986053 0.12603800 0.14663163 0.14985123 0.11146123
## [37] 0.08206938 0.08808615 0.05871973 0.01380575
```

```
# p-value appliqué à lag 40 = 0.013 < 5%, on rejette l'hypothèse nulle
# d'abscence d'autocorrélation pour  rte.
# Pour modéliser cette caractéristique nous utiliserons un modèle ARMA(p,q)
```

Arima

```
reg<-Arima(rte, order=c(2,0,2))
coeftest(reg)
```

ARMA(2,2)

```
## Warning in sqrt(diag(se)): Production de NaN
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ar1        0.06971948         NaN     NaN      NaN
## ar2        0.26419201  0.22531324  1.1726   0.2410
## ma1       -0.08467370         NaN     NaN      NaN
## ma2       -0.27918369  0.17339181 -1.6101   0.1074
## intercept -0.00015419  0.00050320 -0.3064   0.7593
```

```r
# Le modèle ne converge pas donc non valide.
```

```r
# Comme on a observé de l'autocorrélation à l'ordre 32, nous testons un MA(32)

reg<-Arima(rte, order=c(0,0,32))
coeftest(reg)
```

MA(32)

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ma1       -0.01498842  0.02127233 -0.7046  0.48106
## ma2       -0.00577425  0.02121330 -0.2722  0.78547
## ma3       -0.00719452  0.02133513 -0.3372  0.73596
## ma4       -0.02622320  0.02123681 -1.2348  0.21691
## ma5        0.01067383  0.02130427  0.5010  0.61636
## ma6        0.01329528  0.02141187  0.6209  0.53465
## ma7        0.02924908  0.02149980  1.3604  0.17369
## ma8       -0.03495744  0.02148603 -1.6270  0.10374
## ma9        0.00022732  0.02161373  0.0105  0.99161
## ma10      -0.03319557  0.02146839 -1.5463  0.12204
## ma11       0.00926538  0.02164104  0.4281  0.66855
## ma12       0.04234129  0.02155559  1.9643  0.04950 *
## ma13       0.00892586  0.02172125  0.4109  0.68113
## ma14       0.00302889  0.02160615  0.1402  0.88851
## ma15       0.00352187  0.02170974  0.1622  0.87113
## ma16      -0.03077517  0.02162617 -1.4231  0.15472
## ma17      -0.01938695  0.02151274 -0.9012  0.36749
## ma18      -0.00217603  0.02254884 -0.0965  0.92312
## ma19       0.00693063  0.02130148  0.3254  0.74491
## ma20      -0.00912936  0.02277708 -0.4008  0.68856
## ma21      -0.02359477  0.02175152 -1.0847  0.27804
## ma22       0.01571586  0.02225286  0.7062  0.48004
## ma23      -0.01967033  0.02149130 -0.9153  0.36005
## ma24      -0.00129459  0.02192386 -0.0590  0.95291
## ma25      -0.01939906  0.02188130 -0.8866  0.37532
## ma26      -0.04792630  0.02246034 -2.1338  0.03286 *
## ma27       0.02679826  0.02179092  1.2298  0.21878
## ma28       0.01201131  0.02235009  0.5374  0.59098
## ma29      -0.02522966  0.02154360 -1.1711  0.24156
## ma30      -0.03253519  0.02201657 -1.4778  0.13947
## ma31       0.03150606  0.02228388  1.4139  0.15741
## ma32       0.04870758  0.02163074  2.2518  0.02434 *
## intercept -0.00016139  0.00047497 -0.3398  0.73402
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# H0 : theta(k) = 0 VS Ha : theta(k) =/= 0
# Les coefficients ne sont pas tous significatifs.
# On va fixer à 0 des coefficients qui ne sont pas significatifs un à un.
```

```
# # Nous commence par fixer à 0 theta(9) car sa p-value = 0.991 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, NA, NA, NA, NA, NA, NA, NA, 0, NA,
#                                            NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(24) car sa p-value = 0.952 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, NA, NA, NA, NA, NA, NA, NA, 0, NA,
#                                            NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(18) car sa p-value = 0.923 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, NA, NA, NA, NA, NA, NA, NA, 0, NA,
#                                            NA, NA, NA, NA, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(14) car sa p-value = 0.893 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, NA, NA, NA, NA, NA, NA, NA, 0, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(15) car sa p-value = 0.870  > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, NA, NA, NA, NA, NA, NA, NA, 0, NA,
#                                            NA, NA, NA, 0, 0, NA, NA, 0, NA, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(2) car sa p-value = 0.777  > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, 0, NA, NA, NA, NA, NA, NA, 0, NA,
#                                            NA, NA, NA, 0, 0, NA, NA, 0, NA, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(19) car sa p-value = 0.755  > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, 0, NA, NA, NA, NA, NA, NA, 0, NA,
#                                            NA, NA, NA, 0, 0, NA, NA, 0, 0, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
```

```
# # Fixons à 0 theta(3) car sa p-value = 0.730  > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, 0, 0, NA, NA, NA, NA, NA, 0, NA,
#                                            NA, NA, NA, 0, 0, NA, NA, 0, 0, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(20) car sa p-value = 0.700  > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, 0, 0, NA, NA, NA, NA, NA, 0, NA,
#                                            NA, NA, NA, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(11) car sa p-value = 0.668  > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, 0, 0, NA, NA, NA, NA, NA, 0, NA,
#                                            0, NA, NA, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(13) car sa p-value = 0.666  > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, 0, 0, NA, NA, NA, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(5) car sa p-value = 0.609 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, 0, 0, NA, 0, NA, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(28) car sa p-value = 0.605 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, 0, 0, NA, 0, NA, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, NA, NA, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(6) car sa p-value = 0.530 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(NA, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, NA, NA, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(1) car sa p-value = 0.486 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, NA, NA, 0, NA, NA, NA, 0, NA, NA,
```

```
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(22) car sa p-value = 0.476 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, 0, NA, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(23) car sa p-value = 0.358 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, 0, 0, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(17) car sa p-value = 0.375 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, 0, 0, 0, 0,
#                                            NA, 0, 0, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(25) car sa p-value = 0.335 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, 0, 0, 0, 0,
#                                            NA, 0, 0, 0, 0, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(21) car sa p-value = 0.259 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, 0, 0, 0, 0,
#                                            0, 0, 0, 0, 0, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(29) car sa p-value = 0.262 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, 0, 0, 0, 0,
#                                            0, 0, 0, 0, 0, NA, NA, 0, 0, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(4) car sa p-value = 0.243 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, 0, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, 0, 0, 0, 0,
#                                            0, 0, 0, 0, 0, NA, NA, 0, 0, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(16) car sa p-value = 0.192 > 0,05.
```

```r
# reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, 0, 0, 0, NA, NA, 0, NA,
#                                           0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
#                                           0, 0, 0, 0, 0, NA, NA, 0, 0, NA,
#                                           NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(27) car sa p-value = 0.188 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, 0, 0, 0, NA, NA, 0, NA,
#                                           0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
#                                           0, 0, 0, 0, 0, NA, 0, 0, 0, NA,
#                                           NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(7) car sa p-value = 0.151 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, 0, 0, 0, 0, NA, 0, NA,
#                                           0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
#                                           0, 0, 0, 0, 0, NA, 0, 0, 0, NA,
#                                           NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(30) car sa p-value = 0.113 > 0,05.
# reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, 0, 0, 0, 0, NA, 0, NA,
#                                           0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
#                                           0, 0, 0, 0, 0, NA, 0, 0, 0, 0,
#                                           NA, NA, NA))
# coeftest(reg)
```

```r
# Fixons à 0 theta(10) car sa p-value = 0.130 > 0,05.
reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, 0, 0, 0, 0, NA, 0, 0,
                                          0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
                                          0, 0, 0, 0, 0, NA, 0, 0, 0, 0,
                                          NA, NA, NA))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ma8       -0.03657909  0.02186022 -1.6733  0.09426 .
## ma12       0.04263514  0.02107906  2.0226  0.04311 *
## ma26      -0.04396313  0.02178860 -2.0177  0.04362 *
## ma31       0.03666302  0.02177428  1.6838  0.09222 .
## ma32       0.05633863  0.02090546  2.6949  0.00704 **
## intercept -0.00016270  0.00055298 -0.2942  0.76859
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Fixons à 0 theta(8) car sa p-value = 0.094 > 0,05.
reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                                          0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
                                          0, 0, 0, 0, 0, NA, 0, 0, 0, 0,
                                          NA, NA, NA))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ma12       0.04374393  0.02102725  2.0803  0.03749 *
## ma26      -0.04382453  0.02175418 -2.0145  0.04395 *
## ma31       0.03512490  0.02188958  1.6046  0.10857
## ma32       0.05936750  0.02077072  2.8582  0.00426 **
## intercept -0.00016273  0.00057385 -0.2836  0.77673
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Fixons à 0 theta(31) car sa p-value = 0.108 > 0,05.
reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                                           0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0, NA, 0, 0, 0, 0,
                                           0, NA, NA))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ma12       0.04427422  0.02103055  2.1052 0.035271 *
## ma26      -0.04146510  0.02166376 -1.9140 0.055616 .
## ma32       0.05746879  0.02070872  2.7751 0.005518 **
## intercept -0.00016266  0.00055654 -0.2923 0.770086
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Fixons à 0 theta(26) car sa p-value = 0.055 > 0,05.
reg<-Arima(rte, order=c(0,0,32), fixed = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                                           0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                                           0, NA, NA))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ma12       0.04306772  0.02103476  2.0475 0.040613 *
## ma32       0.05606903  0.02073492  2.7041 0.006849 **
## intercept -0.00016317  0.00057716 -0.2827 0.777400
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Enlevons la constante car elle n'est pas significatif (p-value = 0.777 > 0,05).
reg1<-Arima(rte, order=c(0,0,32),
           include.mean = F,
           fixed = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                     0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
                     0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                     0, NA),)
coeftest(reg1)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma12 0.043073   0.021034  2.0477 0.040587 *
## ma32 0.056104   0.020735  2.7058 0.006814 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Maintenant que nous avons un modele avec tous les coefficients qui sont significatif
# Celui-ci n'est valide que si les aléas sont des Bruits Blancs i.e.
# E[epsilon] = 0 et epsilon ne sont pas autocorrélés

# Test de l'espérance nulle des erreurs
# H0 : E[epsilon] = 0 VS Ha : E[epsilon] =/= 0
residu<-reg1$res
t.test(residu)
```

```
##
##  One Sample t-test
##
## data:  residu
## t = -0.27562, df = 2212, p-value = 0.7829
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.001175058  0.000885459
## sample estimates:
##     mean of x
## -0.0001447994
```

```
# p-value = 0.7829 > 0.05 donc on accepte H0
# L'espérance des aléas sont nulles
```

```
library(tseries)
```

```
residuv=(residu-mean(residu))/sd(residu)
```

AR(32) Pour la même raison que nous avons tester le MA(32), nous testons un AR(32)

```
reg<-Arima(rte, order=c(32,0,0))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
```

```
## ar1       -0.01717394  0.02121745 -0.8094 0.418271
## ar2       -0.00567751  0.02120838 -0.2677 0.788929
## ar3       -0.00235299  0.02119439 -0.1110 0.911601
## ar4       -0.02828958  0.02118082 -1.3356 0.181673
## ar5        0.00749994  0.02118806  0.3540 0.723361
## ar6        0.01843443  0.02118374  0.8702 0.384182
## ar7        0.03702899  0.02116979  1.7491 0.080266 .
## ar8       -0.04123031  0.02118221 -1.9465 0.051600 .
## ar9       -0.01344167  0.02119833 -0.6341 0.526021
## ar10      -0.02638681  0.02120108 -1.2446 0.213280
## ar11       0.01521122  0.02120604  0.7173 0.473185
## ar12       0.03464006  0.02121597  1.6327 0.102525
## ar13      -0.00164611  0.02125325 -0.0775 0.938264
## ar14       0.00260350  0.02124520  0.1225 0.902467
## ar15       0.01814729  0.02127408  0.8530 0.393646
## ar16      -0.02822247  0.02126176 -1.3274 0.184382
## ar17      -0.02132239  0.02127418 -1.0023 0.316215
## ar18      -0.00864758  0.02127070 -0.4065 0.684339
## ar19       0.00912711  0.02127110  0.4291 0.667861
## ar20      -0.00759527  0.02128068 -0.3569 0.721160
## ar21      -0.02725573  0.02127832 -1.2809 0.200223
## ar22       0.01701373  0.02128810  0.7992 0.424167
## ar23      -0.01181717  0.02128959 -0.5551 0.578848
## ar24      -0.00035810  0.02128658 -0.0168 0.986578
## ar25      -0.01946321  0.02126535 -0.9153 0.360058
## ar26      -0.04050762  0.02125386 -1.9059 0.056664 .
## ar27       0.02866488  0.02126889  1.3477 0.177743
## ar28       0.01264354  0.02131638  0.5931 0.553089
## ar29      -0.03289432  0.02130746 -1.5438 0.122638
## ar30      -0.02932632  0.02131457 -1.3759 0.168858
## ar31       0.02906439  0.02131716  1.3634 0.172748
## ar32       0.06036192  0.02134530  2.8279 0.004686 **
## intercept -0.00016177  0.00048659 -0.3325 0.739543
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# H0 : phi_k = 0 VS Ha : phi_k =/= 0
# Les coefficients ne sont pas tous significatifs.
# On va fixer à 0 des coefficients qui ne sont pas significatifs un à un.
```

```
# # Nous commence par fixer à 0 phi(24) car sa p-value = 0.986 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(13) car sa p-value = 0.937 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, 0, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
```

```
# # Fixons à 0 theta(3) car sa p-value = 0.913 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, NA, 0, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, 0, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(14) car sa p-value = 0.902 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, NA, 0, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, 0, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(2) car sa p-value = 0.792 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, 0, 0, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, 0, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(18) car sa p-value = 0.685 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, 0, 0, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(20) car sa p-value = 0.728 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, 0, 0, NA, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, NA, 0,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(5) car sa p-value = 0.723 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, 0, 0, NA, 0, NA, NA, NA, NA, NA,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, NA, 0,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(19) car sa p-value = 0.653 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, 0, 0, NA, 0, NA, NA, NA, NA, NA,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, 0, 0,
#                                            NA, NA, NA, 0, NA, NA, NA, NA, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(23) car sa p-value = 0.570 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, 0, 0, NA, 0, NA, NA, NA, NA, NA,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, 0, 0,
#                                            NA, NA, 0, 0, NA, NA, NA, NA, NA, NA,
```

```
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(28) car sa p-value = 0.537 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, 0, 0, NA, 0, NA, NA, NA, NA, NA,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, 0, 0,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(9) car sa p-value = 0.509 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, 0, 0, NA, 0, NA, NA, NA, 0, NA,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, 0, 0,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(11) car sa p-value = 0.496 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(NA, 0, 0, NA, 0, NA, NA, NA, 0, NA,
#                                            0, NA, 0, 0, NA, NA, NA, 0, 0, 0,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(1) car sa p-value = 0.424 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, NA, 0, NA, NA, NA, 0, NA,
#                                            0, NA, 0, 0, NA, NA, NA, 0, 0, 0,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(15) car sa p-value = 0.412 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, NA, 0, NA, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(6) car sa p-value = 0.386 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, NA, 0, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(22) car sa p-value = 0.392 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, 0, 0, 0, NA, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(25) car sa p-value = 0.357 > 0,05.
```

```
# reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, NA, 0, 0, 0,
#                                            NA, 0, 0, 0, 0, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(17) car sa p-value = 0.376 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, 0, 0, 0, 0,
#                                            NA, 0, 0, 0, 0, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(21) car sa p-value = 0.217 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, NA,
#                                            0, NA, 0, 0, 0, NA, 0, 0, 0, 0,
#                                            0, 0, 0, 0, 0, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(10) car sa p-value = 0.209 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, NA, 0, 0, NA, NA, 0, 0,
#                                            0, NA, 0, 0, 0, NA, 0, 0, 0, 0,
#                                            0, 0, 0, 0, 0, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(4) car sa p-value = 0.200 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, 0, 0, 0, NA, NA, 0, 0,
#                                            0, NA, 0, 0, 0, NA, 0, 0, 0, 0,
#                                            0, 0, 0, 0, 0, NA, NA, 0, NA, NA,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(30) car sa p-value = 0.178 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, 0, 0, 0, NA, NA, 0, 0,
#                                            0, NA, 0, 0, 0, NA, 0, 0, 0, 0,
#                                            0, 0, 0, 0, 0, NA, NA, 0, NA, 0,
#                                            NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(31) car sa p-value = 0.159 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, 0, 0, 0, NA, NA, 0, 0,
#                                            0, NA, 0, 0, 0, NA, 0, 0, 0, 0,
#                                            0, 0, 0, 0, 0, NA, NA, 0, NA, 0,
#                                            0, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(16) car sa p-value = 0.161 > 0,05.
# reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, 0, 0, 0, NA, NA, 0, 0,
#                                            0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
#                                            0, 0, 0, 0, 0, NA, NA, 0, NA, 0,
#                                            0, NA, NA))
```

```r
# coeftest(reg)
```

```r
# Fixons à 0 theta(27) car sa p-value = 0.158 > 0,05.
reg<-Arima(rte, order=c(32,0,0), fixed = c(0, 0, 0, 0, 0, 0, NA, NA, 0, 0,
                                           0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0, NA, 0, 0, NA, 0,
                                           0, NA, NA))
coeftest(reg)
```

```
## Warning in sqrt(diag(se)): Production de NaN
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value  Pr(>|z|)
## ar7        3.8008e-02  7.9609e-03  4.7744 1.802e-06 ***
## ar8       -3.9989e-02         NaN     NaN       NaN
## ar12       3.7773e-02         NaN     NaN       NaN
## ar26      -3.6720e-02         NaN     NaN       NaN
## ar29      -3.2751e-02  1.1585e-02 -2.8270  0.004699 **
## ar32       5.7327e-02  2.6478e-02  2.1651  0.030383 *
## intercept -8.7441e-05  5.3691e-04 -0.1629  0.870629
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Le modèle ne converge pas donc non valide.
```

```r
reg<-Arima(rte, order=c(1,0,1))
coeftest(reg)
```

ARMA(1,1)

```
## Warning in sqrt(diag(se)): Production de NaN
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ar1       -0.00761520         NaN     NaN      NaN
## ma1       -0.00837310         NaN     NaN      NaN
## intercept -0.00016222  0.00051855 -0.3128   0.7544
```

```r
# Le modèle ne converge pas donc non valide.
```

PROPRIETE 4 : Clusters de volatilité

```
# Nous analysons maintenant l'existence de clusters de volatilité.
# Ce phénomène, remet en cause l’hypothèse d’homoscédasticité conditionnelle.
# Afin de vérifier formellement si la variance dépend des chocs passés, nous
# appliquons le test ARCH d’Engle (1982) sur la série  rte.
```

```
library(FinTS)
```

```
LM1<-ArchTest(as.numeric(rte),lag=1)
LM1 # p-value < 2.2e-16 < 5%
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  as.numeric(rte)
## Chi-squared = 23.644, df = 1, p-value = 1.159e-06
```

```
# On rejette l’hypothèse nulle d'homoscédasticité conditionnelle donc présence de
# clusters de volatilités
```

PROPRIETE 5 : Queues épaisses conditionnelles

```
library(tseries)
library(fGarch)
library(FinTS)
library(moments)
```

```
# Après avoir estimé l’équation de la moyenne avec un modèle MA(25) et vérifié que
# ses résidus ne présentent plus d’autocorrélation, le test d’Engle met en évidence
# la présence de volatilité conditionnelle. Il est donc nécessaire de modéliser la
# variance avec un modèle GARCH, en particulier le GARCH(1,1).

volat<-garch(residuv,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  1.034e+03
##      1    3  9.944e+02  3.83e-02  7.46e-02  5.5e-02  7.7e+03  1.0e-01  2.88e+02
```

```
##     2    5  9.916e+02  2.84e-03  2.88e-03  3.7e-03  4.9e+01  1.0e-02  4.41e+01
##     3    7  9.864e+02  5.26e-03  5.27e-03  7.4e-03  3.4e+00  2.0e-02  1.07e+00
##     4    9  9.854e+02  9.87e-04  9.87e-04  1.5e-03  4.4e+01  4.0e-03  9.32e-01
##     5   11  9.836e+02  1.90e-03  1.90e-03  3.0e-03  6.1e+00  8.0e-03  7.98e-01
##     6   13  9.801e+02  3.53e-03  3.53e-03  6.2e-03  3.4e+00  1.6e-02  7.56e-01
##     7   15  9.794e+02  6.61e-04  6.61e-04  1.2e-03  4.4e+01  3.2e-03  6.72e-01
##     8   17  9.793e+02  1.30e-04  1.30e-04  2.5e-04  2.1e+02  6.4e-04  5.78e-01
##     9   19  9.791e+02  2.59e-04  2.59e-04  5.0e-04  2.7e+01  1.3e-03  5.60e-01
##    10   21  9.790e+02  5.15e-05  5.15e-05  1.0e-04  5.2e+02  2.6e-04  5.54e-01
##    11   23  9.789e+02  1.03e-04  1.03e-04  2.0e-04  6.5e+01  5.1e-04  5.47e-01
##    12   25  9.787e+02  2.04e-04  2.04e-04  4.0e-04  3.3e+01  1.0e-03  5.44e-01
##    13   28  9.787e+02  4.07e-06  4.07e-06  8.0e-06  6.3e+03  2.0e-05  5.39e-01
##    14   30  9.787e+02  8.14e-07  8.14e-07  1.6e-06  3.2e+04  4.1e-06  5.34e-01
##    15   32  9.787e+02  1.63e-06  1.63e-06  3.2e-06  4.0e+03  8.2e-06  5.34e-01
##    16   34  9.787e+02  3.26e-07  3.26e-07  6.4e-07  7.9e+04  1.6e-06  5.34e-01
##    17   36  9.787e+02  6.51e-07  6.51e-07  1.3e-06  9.9e+03  3.3e-06  5.34e-01
##    18   38  9.787e+02  1.30e-07  1.30e-07  2.6e-07  2.0e+05  6.6e-07  5.34e-01
##    19   40  9.787e+02  2.60e-07  2.60e-07  5.1e-07  2.5e+04  1.3e-06  5.34e-01
##    20   42  9.787e+02  5.21e-08  5.21e-08  1.0e-07  5.0e+05  2.6e-07  5.34e-01
##    21   44  9.787e+02  1.04e-07  1.04e-07  2.1e-07  6.2e+04  5.2e-07  5.34e-01
##    22   46  9.787e+02  2.08e-08  2.08e-08  4.1e-08  1.2e+06  1.0e-07  5.34e-01
##    23   48  9.787e+02  4.17e-08  4.17e-08  8.2e-08  1.5e+05  2.1e-07  5.34e-01
##    24   50  9.787e+02  8.33e-09  8.33e-09  1.6e-08  3.1e+06  4.2e-08  5.34e-01
##    25   52  9.787e+02  1.67e-08  1.67e-08  3.3e-08  3.9e+05  8.4e-08  5.34e-01
##    26   54  9.787e+02  3.33e-09  3.33e-09  6.6e-09  7.7e+06  1.7e-08  5.34e-01
##    27   56  9.787e+02  6.67e-09  6.67e-09  1.3e-08  9.7e+05  3.4e-08  5.34e-01
##    28   58  9.787e+02  1.33e-08  1.33e-08  2.6e-08  4.8e+05  6.7e-08  5.34e-01
##    29   60  9.787e+02  2.67e-09  2.67e-09  5.3e-09  9.7e+06  1.3e-08  5.34e-01
##    30   62  9.787e+02  5.33e-10  5.33e-10  1.1e-09  4.8e+07  2.7e-09  5.34e-01
##    31   64  9.787e+02  1.07e-09  1.07e-09  2.1e-09  6.0e+06  5.4e-09  5.34e-01
##    32   66  9.787e+02  2.13e-10  2.13e-10  4.2e-10  1.2e+08  1.1e-09  5.34e-01
##    33   68  9.787e+02  4.27e-10  4.27e-10  8.4e-10  1.5e+07  2.1e-09  5.34e-01
##    34   70  9.787e+02  8.53e-11  8.53e-11  1.7e-10  1.9e+00  4.3e-10 -4.64e-02
##    35   72  9.787e+02  1.71e-10  1.71e-10  3.4e-10  3.8e+07  8.6e-10  5.34e-01
##    36   74  9.787e+02  3.41e-10  3.41e-10  6.7e-10  1.9e+07  1.7e-09  5.34e-01
##    37   77  9.787e+02  6.83e-12  6.83e-12  1.3e-11  1.9e+00  3.4e-11 -4.64e-02
##    38   79  9.787e+02  1.37e-11  1.37e-11  2.7e-11  1.9e+00  6.9e-11 -4.64e-02
##    39   81  9.787e+02  2.73e-11  2.73e-11  5.4e-11  1.9e+00  1.4e-10 -4.64e-02
##    40   83  9.787e+02  5.46e-12  5.46e-12  1.1e-11  1.9e+00  2.7e-11 -4.64e-02
##    41   85  9.787e+02  1.09e-12  1.09e-12  2.2e-12  1.9e+00  5.5e-12 -4.64e-02
##    42   87  9.787e+02  2.18e-12  2.18e-12  4.3e-12  1.9e+00  1.1e-11 -4.64e-02
##    43   89  9.787e+02  4.37e-12  4.37e-12  8.6e-12  1.9e+00  2.2e-11 -4.64e-02
##    44   92  9.787e+02  8.40e-14  8.74e-14  1.7e-13  1.9e+00  4.4e-13 -4.64e-02
##    45   94  9.787e+02  1.82e-14  1.75e-14  3.4e-14  1.9e+00  8.8e-14 -4.64e-02
##    46   96  9.787e+02  3.44e-14  3.50e-14  6.9e-14  1.9e+00  1.8e-13 -4.64e-02
##    47   98  9.787e+02  7.29e-14  6.99e-14  1.4e-13  1.9e+00  3.5e-13 -4.65e-02
##    48  100  9.787e+02  1.32e-14  1.40e-14  2.8e-14  1.9e+00  7.0e-14 -4.64e-02
##    49  102  9.787e+02  3.48e-15  2.80e-15  5.5e-15  1.8e+00  1.4e-14 -4.57e-02
##    50  104  9.787e+02  5.11e-15  5.59e-15  1.1e-14  1.9e+00  2.8e-14 -4.98e-02
##    51  105  9.787e+02 -1.02e+07  1.12e-14  2.2e-14  1.8e+00  5.6e-14 -4.61e-02
##
##  ***** FALSE CONVERGENCE *****
##
##  FUNCTION     9.786967e+02   RELDX        2.205e-14
```

```
##  FUNC. EVALS      105          GRAD. EVALS       51
##  PRELDF       1.118e-14      NPRELDF     -4.614e-02
##
##      I      FINAL X(I)        D(I)          G(I)
##
##      1    8.432008e-01     1.000e+00     1.284e+02
##      2    1.796234e-01     1.000e+00    -1.025e+02
##      3    1.146777e-14     1.000e+00     1.040e+02
```

```r
# Ne converge pas donc faire garchFit
```

```r
volat<-garchFit(~garch(1,1),data=residuv,trace=F,include.mean = FALSE)
summary(volat)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = residuv, include.mean = FALSE,
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x0000028fdceb7c78>
##  [data = residuv]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##    omega    alpha1     beta1
## 0.060455  0.143038  0.809829
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega    0.06045     0.01156    5.229  1.7e-07 ***
## alpha1   0.14304     0.01690    8.463  < 2e-16 ***
## beta1    0.80983     0.02055   39.410  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  -2967.656    normalized:  -1.341011
##
## Description:
##  Sat Dec  6 14:48:22 2025 by user: celes
##
##
## Standardised Residuals Tests:
```

```
##                                     Statistic   p-Value
##  Jarque-Bera Test   R    Chi^2  6284.8806544 0.0000000
##  Shapiro-Wilk Test  R    W         0.9404922 0.0000000
##  Ljung-Box Test     R    Q(10)     8.7930273 0.5518524
##  Ljung-Box Test     R    Q(15)     9.4134409 0.8549274
##  Ljung-Box Test     R    Q(20)    11.4563329 0.9335195
##  Ljung-Box Test     R^2  Q(10)     7.8111116 0.6472813
##  Ljung-Box Test     R^2  Q(15)     9.8681698 0.8279494
##  Ljung-Box Test     R^2  Q(20)    10.9986061 0.9462587
##  LM Arch Test       R    TR^2      8.7427736 0.7247298
##
## Information Criterion Statistics:
##      AIC      BIC      SIC     HQIC
## 2.684732 2.692462 2.684729 2.687556
```

```r
# HO : alpha(k) = 0 VS Ha : alpha(k) =/= 0
# On a nos coefficients qui sont significatifs (p-value < 5%).

# On souhaite maintenant savoir si avec notre modèle on a réussi à prendre en compte
# toutes l'hétéroscédasticité conditionnelle
```

```r
resvolat=volat@residuals/volat@sigma.t
anscombe.test(resvolat)
```

```
##
##  Anscombe-Glynn kurtosis test
##
## data:  resvolat
## kurt = 11.244, z = 18.810, p-value < 2.2e-16
## alternative hypothesis: kurtosis is not equal to 3
```

```r
# p-value < 2.2e-16 < 5 %, nous rejetons l'hypothèse nulle selon laquelle la kurtosis
# serait égale à 3.
# Ainsi, nous pouvons interpréter sa valeur calculé : kurt = 11.657 >> 3, cela
# indique une distribution fortement leptokurtique


# Même après la prise en compte de la volatilité conditionnelle, les résidus
# présentent donc encore des queues épaisses, confirmant la présence de leptokurtose conditionnelle.
```

PROPRIETE 8 : Stationnarité

```r
library(urca)
```

Test de Dickey-Fuller (DF)

```r
summary(ur.df(rte,type= "trend",lags=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
## -0.35211 -0.01200  0.00019  0.01162  0.16433
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.445e-04  1.054e-03   0.422    0.673
## z.lag.1     -1.016e+00  2.128e-02 -47.749   <2e-16 ***
## tt          -5.504e-07  8.254e-07  -0.667    0.505
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02479 on 2209 degrees of freedom
## Multiple R-squared:  0.5079, Adjusted R-squared:  0.5075
## F-statistic:  1140 on 2 and 2209 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -47.7489 759.9854 1139.978
##
## 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
```

```r
# H0 : B1 = 0 vs Ha : B1 =/= 0
# p-value(B1) = 0.505 > 0.05 le coefficient n'est pas significatif
# Donc on estime drift
```

```r
summary(ur.df(rte,type= "drift",lags=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
## -0.35242 -0.01202  0.00016  0.01132  0.16415
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0001645  0.0005270  -0.312    0.755
## z.lag.1     -1.0158150  0.0212735 -47.750   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02478 on 2210 degrees of freedom
## Multiple R-squared:  0.5078, Adjusted R-squared:  0.5076
## F-statistic:  2280 on 1 and 2210 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -47.7502 1140.042
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

```r
# H0 : B0 = 0 vs Ha : B0 =/= 0
# p-value(B0) = 0.755 > 0.05 B0 n'est pas significatif
# Donc on estime none
```

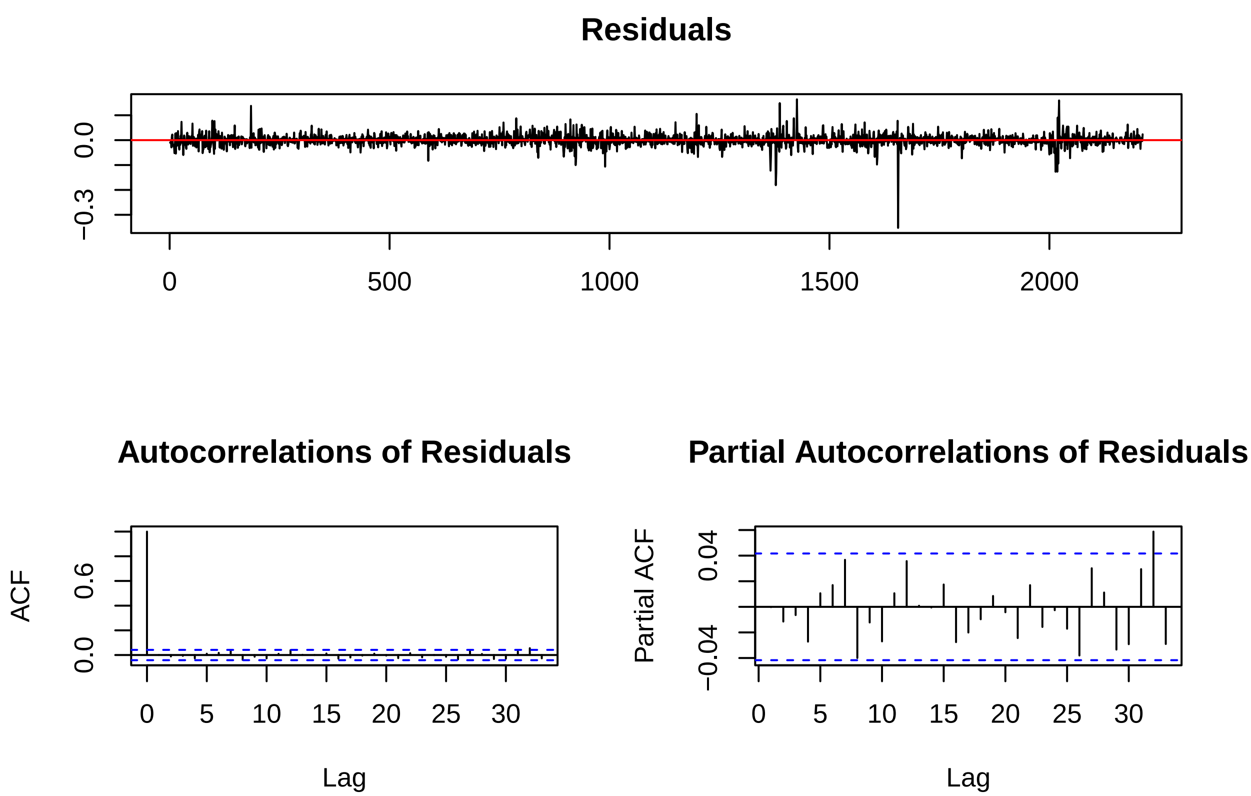
```r
summary(ur.df(rte,type= "drift",lags=0))
```

```
##
## ###############################################
## # 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
## -0.35242 -0.01202  0.00016  0.01132  0.16415
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0001645  0.0005270  -0.312    0.755
## z.lag.1     -1.0158150  0.0212735 -47.750   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02478 on 2210 degrees of freedom
## Multiple R-squared:  0.5078, Adjusted R-squared:  0.5076
## F-statistic:  2280 on 1 and 2210 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -47.7502 1140.042
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

```
# H0 : B0 = 0 vs Ha : B0 =/= 0
# p-value(B0) = 0.755 > 0.05 B0 n'est pas significatif
# Donc on estime none
```

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




```
# La figure nous indique que les aléas sont auto-corrélés aux ordres 32 et donc
# notre conclusion concernant l'absence de RU n'est pas valide.
# Nous devons effectuer un test de RU dans le cadre de la régression de Dickey Fuller Augmenté.
```

Test de Dickey Fuller Augmenté (ADF)

```
library(CADFtest)
```

```
T = length(rte)
Schwert<-as.integer(12*(T/100)^(0.25)) # = 26
```

```
# Nous vérifions que B0 est bien non significatif
summary(CADFtest(rte, criterion="MAIC",type="drift",max.lag.y=Schwert))
```

```
## Augmented DF test
##                                                 ADF test
## t-test statistic:                           -1.708755e+01
## p-value:                                     2.477047e-39
## Max lag of the diff. dependent variable:     6.000000e+00
##
## Call:
## dynlm(formula = formula(model), start = obs.1, end = obs.T)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35383 -0.01202  0.00016  0.01142  0.16085
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.211e-05  5.300e-04  -0.117   0.9067
## L(y, 1)     -1.001e+00  5.858e-02 -17.088   <2e-16 ***
## L(d(y), 1)  -1.434e-02  5.419e-02  -0.265   0.7914
## L(d(y), 2)  -3.039e-02  4.927e-02  -0.617   0.5375
## L(d(y), 3)  -3.479e-02  4.367e-02  -0.797   0.4257
## L(d(y), 4)  -6.440e-02  3.762e-02  -1.712   0.0870 .
## L(d(y), 5)  -5.122e-02  3.049e-02  -1.680   0.0932 .
## L(d(y), 6)  -3.651e-02  2.141e-02  -1.705   0.0883 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02478 on 2178 degrees of freedom
## Multiple R-squared:  0.5086, Adjusted R-squared:  0.507
## F-statistic: 1.045 on 6 and 2178 DF,  p-value: 0.3941
```

```
# Le MAIC nous donne un lag de 6
summary(ur.df(rte,type= "drift",lags=6))
```

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

```
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35344 -0.01203  0.00014  0.01156  0.16084
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0001529  0.0005281  -0.290   0.7722
## z.lag.1     -0.9947357  0.0581883 -17.095   <2e-16 ***
## z.diff.lag1 -0.0222106  0.0538574  -0.412   0.6801
## z.diff.lag2 -0.0336159  0.0489776  -0.686   0.4926
## z.diff.lag3 -0.0390377  0.0434451  -0.899   0.3690
## z.diff.lag4 -0.0657032  0.0374468  -1.755   0.0795 .
## z.diff.lag5 -0.0549148  0.0303869  -1.807   0.0709 .
## z.diff.lag6 -0.0377004  0.0213167  -1.769   0.0771 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0248 on 2198 degrees of freedom
## Multiple R-squared:  0.5093, Adjusted R-squared:  0.5078
## F-statistic: 325.9 on 7 and 2198 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -17.0951 146.1217
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

```r
# p-value(B0) = 0.772 > 0.05 B0 n'est pas significatif
```

```r
# Donc on estime bien none
summary(CADFtest(rte, criterion="MAIC",type="none",max.lag.y=Schwert))
```

```
## Augmented DF test
##                                                  ADF test
## t-test statistic:                           -1.709104e+01
## p-value:                                     4.290017e-36
## Max lag of the diff. dependent variable:    6.000000e+00
##
## Call:
## dynlm(formula = formula(model), start = obs.1, end = obs.T)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35389 -0.01208  0.00009  0.01136  0.16079
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## L(y, 1)    -1.00090    0.05856 -17.091   <2e-16 ***
```

```
## L(d(y), 1) -0.01439    0.05418  -0.266   0.7906
## L(d(y), 2) -0.03043    0.04926  -0.618   0.5368
## L(d(y), 3) -0.03483    0.04366  -0.798   0.4251
## L(d(y), 4) -0.06443    0.03761  -1.713   0.0868 .
## L(d(y), 5) -0.05123    0.03049  -1.681   0.0930 .
## L(d(y), 6) -0.03652    0.02140  -1.706   0.0881 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02477 on 2179 degrees of freedom
## Multiple R-squared:  0.5086, Adjusted R-squared:  0.507
## F-statistic: 1.045 on 6 and 2179 DF,  p-value: 0.3937
```

```
# Le MAIC nous donne un lag de 6
```

```
summary(ur.df(rte,type= "none",lags=6))
```

```
##
## ###############################################
## # 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
## -0.35360 -0.01219 -0.00002  0.01141  0.16068
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -0.99443    0.05817 -17.096   <2e-16 ***
## z.diff.lag1 -0.02247    0.05384  -0.417   0.6764
## z.diff.lag2 -0.03384    0.04896  -0.691   0.4896
## z.diff.lag3 -0.03922    0.04343  -0.903   0.3667
## z.diff.lag4 -0.06584    0.03744  -1.759   0.0788 .
## z.diff.lag5 -0.05500    0.03038  -1.810   0.0704 .
## z.diff.lag6 -0.03774    0.02131  -1.771   0.0767 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0248 on 2199 degrees of freedom
## Multiple R-squared:  0.5093, Adjusted R-squared:  0.5077
## F-statistic: 326.1 on 7 and 2199 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -17.0962
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```r
# On valide la spécification lorsque le dernier gamma est significatif.
# t.stat = |-1.771| > 1.64 donc dernier gamma significatif

# On peut donc passer au test de racine unitaire :
# H0 : rho-1 = 0 vs Ha : rho-1 =/= 0
# On a t.calculé = -17.0962  < -1.95 donc on rejette H0

# Ainsi on peut conclure à partir du test de DFA que le PGD est stationnaire.
```

Test de Zivot et Andrews (ZA)

```r
# Cependant, ces résultat non sont valide que si pendant toute la période d'étude,
# il n'y a pas de choc structurelle u conjoncturelle
# H0 : DS sans date de rupture
# Ha : TS avec une date de rupture
```

```r
# On commence avec comme choix du model "both"
summary(ur.za(rte, model = "both", lag = Schwert))
```

```
##
## ################################
## # Zivot-Andrews Unit Root Test #
## ################################
##
##
## Call:
## lm(formula = testmat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35353 -0.01200  0.00004  0.01148  0.16299
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.396e-03  1.795e-03  -1.892 0.058654 .
## y.l1        -3.173e-01  1.260e-01  -2.518 0.011885 *
## trend        1.120e-05  3.531e-06   3.172 0.001538 **
## y.dl1        2.959e-01  1.229e-01   2.407 0.016151 *
## y.dl2        2.735e-01  1.199e-01   2.282 0.022613 *
## y.dl3        2.629e-01  1.169e-01   2.250 0.024568 *
## y.dl4        2.273e-01  1.138e-01   1.996 0.046024 *
## y.dl5        2.309e-01  1.108e-01   2.084 0.037288 *
## y.dl6        2.370e-01  1.077e-01   2.201 0.027867 *
## y.dl7        2.643e-01  1.047e-01   2.525 0.011636 *
## y.dl8        2.142e-01  1.016e-01   2.107 0.035212 *
## y.dl9        1.961e-01  9.848e-02   1.991 0.046565 *
## y.dl10       1.584e-01  9.528e-02   1.662 0.096582 .
## y.dl11       1.641e-01  9.210e-02   1.782 0.074931 .
## y.dl12       1.911e-01  8.901e-02   2.147 0.031882 *
## y.dl13       1.830e-01  8.579e-02   2.133 0.032999 *
## y.dl14       1.777e-01  8.247e-02   2.155 0.031299 *
```

```
## y.dl15       1.876e-01  7.906e-02   2.373 0.017708 *
## y.dl16       1.532e-01  7.535e-02   2.034 0.042109 *
## y.dl17       1.257e-01  7.137e-02   1.761 0.078321 .
## y.dl18       1.086e-01  6.739e-02   1.611 0.107394
## y.dl19       1.127e-01  6.330e-02   1.780 0.075183 .
## y.dl20       1.008e-01  5.932e-02   1.700 0.089357 .
## y.dl21       6.917e-02  5.479e-02   1.262 0.206919
## y.dl22       7.814e-02  4.973e-02   1.571 0.116223
## y.dl23       5.637e-02  4.406e-02   1.279 0.200912
## y.dl24       4.668e-02  3.790e-02   1.232 0.218249
## y.dl25       2.329e-02  3.068e-02   0.759 0.447820
## y.dl26      -2.215e-02  2.146e-02  -1.032 0.302147
## du          -8.580e-03  2.277e-03  -3.768 0.000169 ***
## dt          -1.015e-05  3.949e-06  -2.570 0.010230 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0247 on 2155 degrees of freedom
##   (27 observations effacées parce que manquantes)
## Multiple R-squared:  0.02024,    Adjusted R-squared:  0.006605
## F-statistic: 1.484 on 30 and 2155 DF,  p-value: 0.04424
##
##
## Teststatistic: -10.4519
## Critical values: 0.01= -5.57 0.05= -5.08 0.1= -4.82
##
## Potential break point at position: 890
```

```
# Dernier gamma non significatif car sa statistique t associée (1.032) est inférieure en valeur absolue
```

```
summary(ur.za(rte, model = "both", lag = 25))
```

```
##
## ################################
## # Zivot-Andrews Unit Root Test #
## ################################
##
##
## Call:
## lm(formula = testmat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35250 -0.01208  0.00008  0.01147  0.16279
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.445e-03  1.789e-03  -1.925 0.054303 .
## y.l1        -3.479e-01  1.227e-01  -2.835 0.004629 **
## trend        1.139e-05  3.518e-06   3.239 0.001220 **
## y.dl1        3.256e-01  1.197e-01   2.719 0.006598 **
## y.dl2        3.025e-01  1.168e-01   2.591 0.009634 **
## y.dl3        2.918e-01  1.137e-01   2.566 0.010355 *
```

```
## y.dl4        2.554e-01  1.108e-01   2.306 0.021206 *
## y.dl5        2.592e-01  1.076e-01   2.408 0.016106 *
## y.dl6        2.645e-01  1.046e-01   2.529 0.011497 *
## y.dl7        2.918e-01  1.015e-01   2.876 0.004073 **
## y.dl8        2.416e-01  9.835e-02   2.456 0.014115 *
## y.dl9        2.232e-01  9.516e-02   2.346 0.019088 *
## y.dl10       1.847e-01  9.203e-02   2.007 0.044874 *
## y.dl11       1.899e-01  8.892e-02   2.135 0.032872 *
## y.dl12       2.168e-01  8.568e-02   2.530 0.011482 *
## y.dl13       2.088e-01  8.235e-02   2.535 0.011321 *
## y.dl14       2.031e-01  7.894e-02   2.573 0.010161 *
## y.dl15       2.133e-01  7.520e-02   2.837 0.004597 **
## y.dl16       1.789e-01  7.125e-02   2.511 0.012116 *
## y.dl17       1.505e-01  6.730e-02   2.236 0.025462 *
## y.dl18       1.328e-01  6.322e-02   2.101 0.035794 *
## y.dl19       1.358e-01  5.923e-02   2.292 0.021982 *
## y.dl20       1.247e-01  5.470e-02   2.280 0.022696 *
## y.dl21       9.336e-02  4.966e-02   1.880 0.060281 .
## y.dl22       1.022e-01  4.399e-02   2.324 0.020227 *
## y.dl23       7.974e-02  3.785e-02   2.107 0.035273 *
## y.dl24       6.980e-02  3.063e-02   2.279 0.022775 *
## y.dl25       4.594e-02  2.143e-02   2.144 0.032157 *
## du          -8.729e-03  2.271e-03  -3.844 0.000124 ***
## dt          -1.035e-05  3.937e-06  -2.628 0.008653 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02469 on 2157 degrees of freedom
##   (26 observations effacées parce que manquantes)
## Multiple R-squared:  0.01976,    Adjusted R-squared:  0.006584
## F-statistic:   1.5 on 29 and 2157 DF,  p-value: 0.04248
##
##
## Teststatistic: -10.9818
## Critical values: 0.01= -5.57 0.05= -5.08 0.1= -4.82
##
## Potential break point at position: 890
```

```
# Dernier gamma significatif car sa statistique t associée (2.144) est supérieure en valeur absolue au

# On observe que delta 1 et delta 2 sont significatif.
# En effet, leurs p-values, respectivement 0.0001 et 0.0086, sont bien
# inférieur à notre seuil de 5%
# De même pour beta(1) avec sa p-value de 0.0012 < 5%.

# On a donc le bon modèle, on peut donc faire le test de racine unitaire :
# H0 : rho-1 = 0
# t.stat = -10.9818 < -5.08 = t.critique ainsi on rejette H0
# Le PGD est donc TS avec date de rupture.
```

```
break_point = 890
dates_rte[break_point] # 2015-08-10
```

```
## [1] "2015-08-10"
```

```r
# Cependant, comme le rendement n'a pas de tendance alors il ne peut pas être TS,
# le PGD est donc stationnaire avec une date de rupture autour du 10 août 2015.
```

Test de Lee et Strazicich (LS)

```r
# Cependant, nous pouvons rejeter/accepter H0 dans ZA alors que possibilité
# d'un PGD DS avec une date de rupture
# On va donc faire un test LS

# Comme dans ZA notre model est both alors dans LS notre model est break

# sourcer LeeStrazicichUnitRootTest.R
myLS_test <- ur.ls(y = rte, model = "break", breaks = 1,
                   lags = Schwert , method = "GTOS", pn = 0.1, print.results = "print")
```

```
## [1] -13.99145
## [1] "First possible structural break at position: 834"
## [1] "The location of the first break - lambda_1: 0.4 , with the number of total observations: 2213"
## Critical values - Break model:
##      lambda    1%    5%   10%
## [1,]    0.1 -5.11 -4.50 -4.21
## [2,]    0.2 -5.07 -4.47 -4.20
## [3,]    0.3 -5.15 -4.45 -4.18
## [4,]    0.4 -5.05 -4.50 -4.18
## [5,]    0.5 -5.11 -4.51 -4.17
## [1] "Number of lags determined by general-to-specific lag selection: 11"
## Runtime:
## Time difference of 0.3616534 mins
```

```r
# t.stat = -13.99145 < -4.5  = t.critique donc on rejette H0
# Ainsi, le PGD qui a généré les données est TS avec une date de rupture.

# Nous avons suffisamment d'observation pour ne pas faire bootstrap (2213)
```

```r
break_point = 834
dates_rte[break_point] # 2020-12-30
```

```
## [1] "2015-05-21"
```

```r
# Cette date coïncide avec la signature d’un contrat majeur de production de
# KUH-1 Surion avec l’agence d’armement DAPA (environ 1 000 milliards de wons).

# Cependant, comme le rendement n'a pas de tendance alors il ne peut pas être TS,
# le PGD est donc stationnaire avec une date de rupture autour du 30 décembre 2020.
```

II - Analyse des 8 propriétés statistiques du rendement logarithmique de la série rtt

```
s=sd(rtt) # écart type estimé
rbar<-mean(rtt) # moyenne empirique
rbar # = 0.00110
```

```
## [1] 0.001106093
```

```
# Est-ce que cette moyenne empirique proche de 0 est statistiquement nulle ?
# H0 : E[rtt] = mu = 0 VS Ha : E[rtt] = mu =/= 0
t.test(rtt)
```

```
##
##  One Sample t-test
##
## data:  rtt
## t = 1.459, df = 1184, p-value = 0.1448
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0003813505  0.0025935369
## sample estimates:
##   mean of x
## 0.001106093
```

```
# Comme p-value = 0.1448 > 0.05, alors on accepte H0.
# Ainsi, l'espérance du PGD qui a généré  rtt est nulle.
```

PROPRIETE 1 - Asymétrie perte/gain

```
library(moments)
```

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

```
##
##  D'Agostino skewness test
##
## data:  rtt
## skew = 0.10391, z = 1.46548, p-value = 0.1428
## alternative hypothesis: data have a skewness
```

```
# p-value = 0.1428 > 5 %, on accepte donc H0 : la skewness n’est pas
# significativement différente de zéro, ce qui indique l’absence d’asymétrie.
```

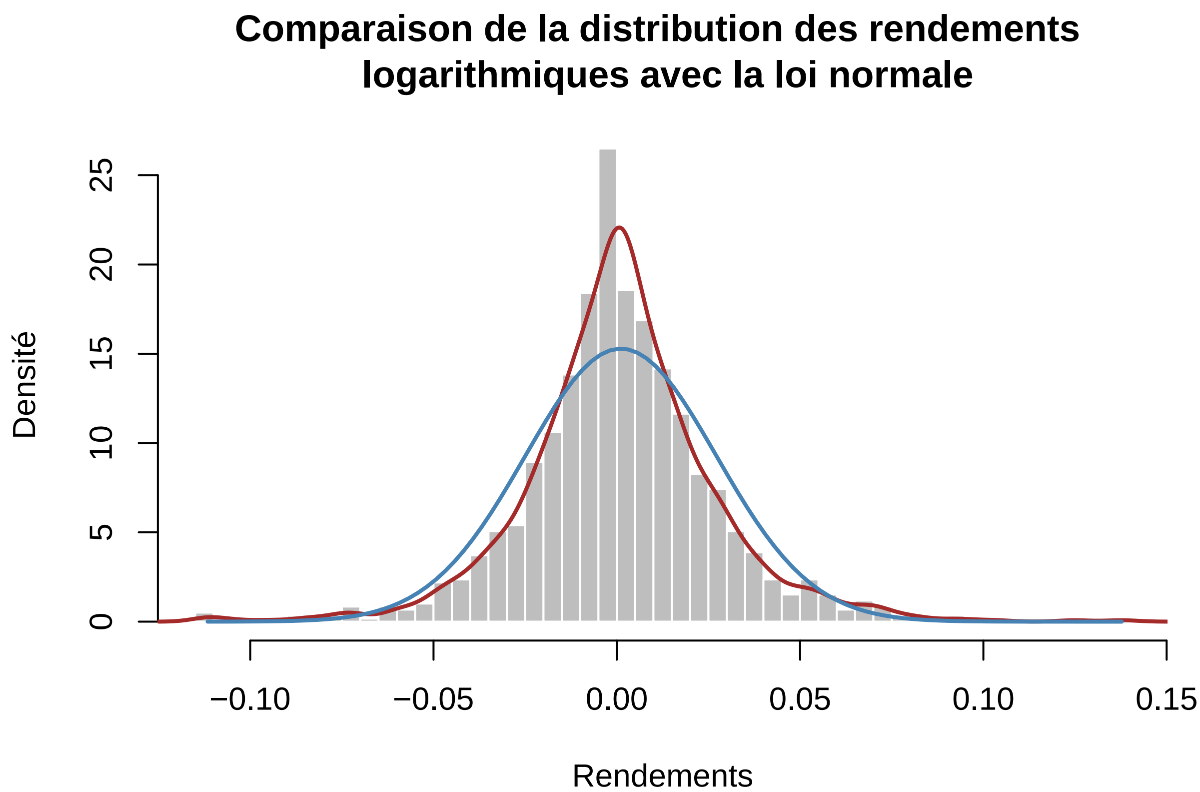
PROPRIETE 2 : Queues de distribution épaisses

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

```
##
##  Anscombe-Glynn kurtosis test
##
## data:  rtt
## kurt = 5.8089, z = 9.1185, p-value < 2.2e-16
## alternative hypothesis: kurtosis is not equal to 3
```

```r
# p-value < 2.2e-16 < 5 %, nous rejetons l'hypothèse nulle selon laquelle la kurtosis
# serait égale à 3.

# Ainsi, nous pouvons interpréter sa valeur calculée : kurt = 5.8089 > 3, cela
# indique une distribution fortement leptokurtique, i.e. une distribution dont la
# cloche est plus pointue que celle de la loi gaussienne, avec des queues plus
# importante.
```

```r
# Comparaison de rtt avec une loi normale :
mu <- mean(rtt, na.rm = TRUE)
sigma <- sd(rtt, na.rm = TRUE)

hist(rtt, breaks = 70, freq = FALSE, col = "grey",
     main = "Comparaison de la distribution des rendements \n logarithmiques avec la loi normale",
     xlab = "Rendements",
     ylab = "Densité",
     border = "white")

lines(density(rtt), col = "brown", lwd = 2)

curve(dnorm(x, mean = mu, sd = sigma),
      from = min(rtt), to = max(rtt),
      add = TRUE, col = "steelblue", lwd = 2)
```




```
# On observe bien la distribution leptokurtique
```

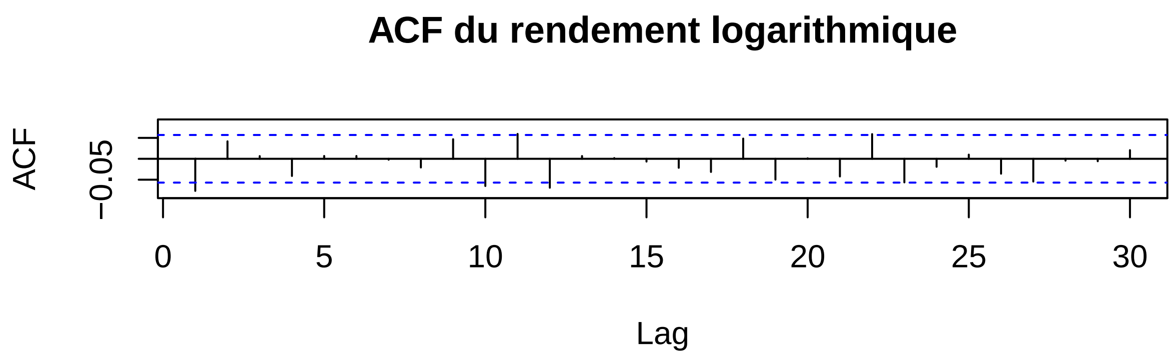
PROPRIETE 3 : Autocorrélations des carrés des rendements forts et faibles pour les rendements

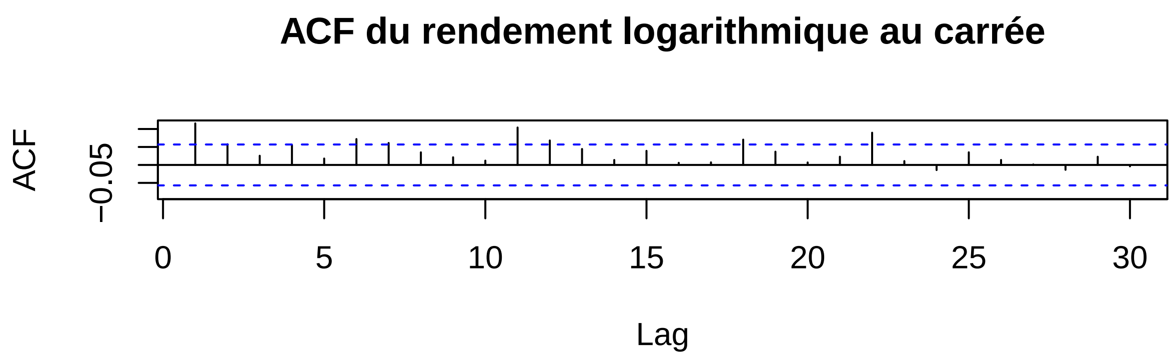
```
library(FinTS)
library(TSA)
library(lmtest)
library(forecast)
```

Analyse de la présence éventuelle d'autocorrélation dans les rendements et dans les rendements au carré

Corrélogramme (ACF)

```
op<-par(mfrow=c(2,1))
Acf(rtt,main='ACF du rendement logarithmique')

Acf(rtt^2,main='ACF du rendement logarithmique au carrée')
```






```r
par(op)
# rho(1), rho(10), rho(11), rho(12), rho(22), rho(23) significatif
# # mais les rho prennent des petite valeur (max 0.07)
# autocorrélation dans les rendements

# autocorrélation aussi dans les rendements au carrées (+ fort)
```

Test de Ljung-Box

```r
# Rendement logarithmique au carrée
Box.test(rtt^2,lag=1,type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  rtt^2
## X-squared = 15.914, df = 1, p-value = 6.629e-05
```

```r
# p-value = 6.629e-05 < 5%
# On rejette donc l'hypothèse nulle d'absence d'autocorrélation
```

```r
# Rendement logarithmique
Box.test(rtt,lag=1,type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  rtt
## X-squared = 7.0002, df = 1, p-value = 0.00815
```

```r
# p-value = 0.00815 < 5%
# On rejette donc l'hypothèse nulle d'absence d'autocorrélation

# Pour modéliser cette caractéristique nous utiliserons un modèle ARMA(p,q)
```

EACF

Dans un premier temps, nous devons déterminer la valeur de p et q du ARMA(p,q) via l'eacf.

```r
eacf(rtt)
```

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

```r
# p = 4 et q = 4
```

Arima

```r
reg<-Arima(rtt, order=c(4,0,4))
coeftest(reg)
```

ARMA(4,4)

```
##
## z test of coefficients:
##
##               Estimate  Std. Error  z value Pr(>|z|)
## ar1        -0.66920323  0.04927771 -13.5802   <2e-16 ***
## ar2         0.25993185  0.02938200   8.8466   <2e-16 ***
## ar3        -0.73052349  0.03285193 -22.2369   <2e-16 ***
```

```
## ar4       -0.88221607  0.04512912 -19.5487   <2e-16 ***
## ma1        0.62411819  0.05976504  10.4429   <2e-16 ***
## ma2       -0.25596985  0.02874961  -8.9034   <2e-16 ***
## ma3        0.76967708  0.02762678  27.8598   <2e-16 ***
## ma4        0.85015261  0.05357741  15.8677   <2e-16 ***
## intercept  0.00103703  0.00074246   1.3967   0.1625
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# H0 : theta(k) = 0 VS Ha : theta(k) =/= 0
# H0 : phi(k) = 0 VS Ha : phi(k) =/= 0

# Les coefficients ne sont pas tous significatifs.
# On va fixer à 0 des coefficients qui ne sont pas significatifs un à un.
```

```
# Nous commence par fixer à 0 phi(2) car sa p-value = 0.8381 > 0,05.
reg<-Arima(rtt, order=c(4,0,4), fixed = c(NA, 0, NA, NA,
                                          NA, NA, NA, NA,
                                          NA))
coeftest(reg)
```

```
## Warning in sqrt(diag(se)): Production de NaN
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error   z value  Pr(>|z|)
## ar1        0.76183731  0.00994266   76.6231 < 2.2e-16 ***
## ar3        0.18545960  0.01681682   11.0282 < 2.2e-16 ***
## ar4       -0.65815849  0.01443245  -45.6027 < 2.2e-16 ***
## ma1       -0.83887476  0.00662594 -126.6046 < 2.2e-16 ***
## ma2        0.09084287  0.02994040    3.0341  0.002412 **
## ma3       -0.19585033  0.00636942  -30.7485 < 2.2e-16 ***
## ma4        0.62783426         NaN       NaN       NaN
## intercept  0.00113311  0.00072526    1.5623  0.118206
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# le modèle ne converge pas, il n'est donc pas valide.
```

```


reg<-Arima(rtt, order=c(1,0,1))
coeftest(reg)
```

ARMA(1,1)

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
```

```
## ar1       -0.32459856  0.22842778 -1.4210   0.1553
## ma1        0.24773924  0.23307140  1.0629   0.2878
## intercept  0.00110706  0.00071208  1.5547   0.1200
```

```
# Fixons à 0 theta(1) car sa p-value = 0.2878 > 0,05.
# Cela revient à faire un AR(1)
```

```
reg<-Arima(rtt, order=c(1,0,0))
coeftest(reg)
```

AR(1)

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ar1       -0.07682307  0.02897800 -2.6511 0.008023 **
## intercept  0.00110529  0.00070232  1.5738 0.115538
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Enlevons la constante car elle n'est pas significatif (p-value = 0.115 > 0,05).
reg1 <-Arima(rtt, order=c(1,0,0), include.mean = F)
coeftest(reg1)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.074915   0.028983 -2.5848 0.009744 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Miantenant que nous avons un modele avec tous les coefficients qui sont significatif
# Celui-ci n'est valide que si les aléas sont des Bruits Blancs i.e.
# E[epsilon] = 0 et epsilon ne sont pas autocorrélés
```

```
# Test de l'espérance nulle des erreurs
# H0 : E[epsilon] = 0 VS Ha : E[epsilon] =/= 0
residu<-reg1$res
t.test(residu)
```

```
##
##  One Sample t-test
##
## data:  residu
## t = 1.5743, df = 1184, p-value = 0.1157
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
```

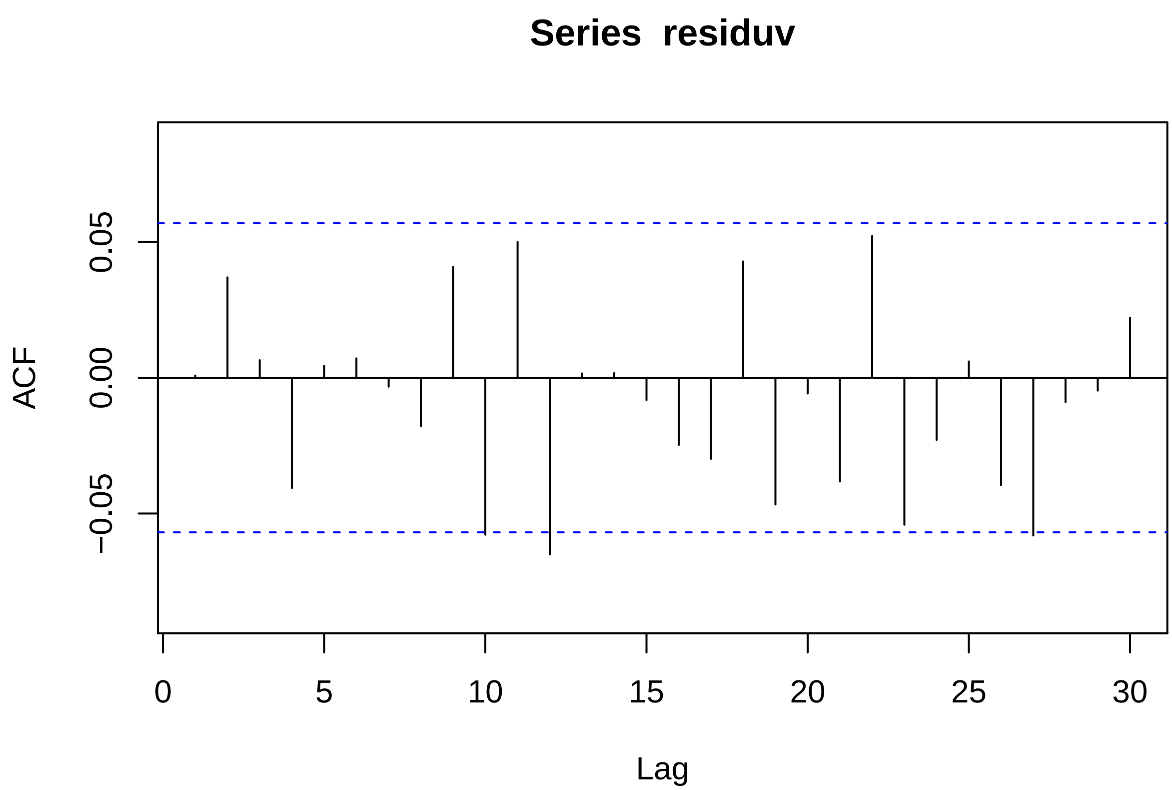
```
##  -0.0002930694  0.0026730213
## sample estimates:
##   mean of x
## 0.001189976
```

```
# p-value = 0.1157 > 0.05 donc on accepte H0
# L'espérance des aléas sont nulles
```

```
# Test d'abscence d'autocorrélation dans les aléas
library(tseries)
```

```
residuv=(residu-mean(residu))/sd(residu)
```

```
Acf(residuv) # On observe encore de l'autocorrélation dans nos résidus
```




```
# rho(10), rho(12) et rho(27) significatif

# Notre modèle n'est donc pas valide.

# Il faut donc un modèle avec P ou Q plus grand.
```

MA(23) Comme on a observé de l'autocorrélation à l'ordre 23 dans l'ACF du rendement logarithmique, nous testons un MA(23)

```r
reg<-Arima(rtt, order=c(0,0,23))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ma1       -0.06302076  0.02903363 -2.1706  0.02996 *
## ma2        0.04029798  0.02912449  1.3836  0.16647
## ma3       -0.00230016  0.02945413 -0.0781  0.93775
## ma4       -0.05674898  0.02959305 -1.9176  0.05516 .
## ma5        0.01409658  0.02959647  0.4763  0.63387
## ma6        0.00741614  0.02953134  0.2511  0.80172
## ma7       -0.01636916  0.02940887 -0.5566  0.57780
## ma8       -0.01779670  0.02941560 -0.6050  0.54517
## ma9        0.03470145  0.02977630  1.1654  0.24386
## ma10      -0.06090813  0.02976556 -2.0463  0.04073 *
## ma11       0.04578963  0.02975870  1.5387  0.12388
## ma12      -0.07368174  0.03060824 -2.4073  0.01607 *
## ma13       0.00019013  0.03141865  0.0061  0.99517
## ma14       0.00362966  0.03028240  0.1199  0.90459
## ma15      -0.01494471  0.02907331 -0.5140  0.60723
## ma16      -0.02029012  0.03107674 -0.6529  0.51382
## ma17      -0.04303108  0.03081917 -1.3962  0.16264
## ma18       0.06405691  0.03007006  2.1303  0.03315 *
## ma19      -0.05452537  0.03081606 -1.7694  0.07683 .
## ma20      -0.01295079  0.03113890 -0.4159  0.67748
## ma21      -0.03045590  0.02984816 -1.0204  0.30756
## ma22       0.05636980  0.02929931  1.9239  0.05436 .
## ma23      -0.07154415  0.03250784 -2.2008  0.02775 *
## intercept  0.00108621  0.00054481  1.9937  0.04618 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# H0 : theta(k) = 0 VS Ha : theta(k) =/= 0
# Les coefficients ne sont pas tous significatifs.
# On va fixer à 0 des coefficients qui ne sont pas significatifs un à un.
```

```r
# # Nous commence par fixer à 0 tous les theta ayant une p-value très élevée :
# # theta(3), theta(5), theta(6), theta(7), theta(8), theta(13), theta(14),
# # theta(15), theta(16), theta(20)
# reg<-Arima(rtt, order=c(0,0,23), fixed = c(NA, NA, 0, NA, 0, 0, 0, 0, NA, NA,
#                                            NA, NA, 0, 0, 0, 0, NA, NA, NA, 0,
#                                            NA, NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(21) car sa p-value = 0.23672 > 0,05.
# reg<-Arima(rtt, order=c(0,0,23), fixed = c(NA, NA, 0, NA, 0, 0, 0, 0, NA, NA,
#                                            NA, NA, 0, 0, 0, 0, NA, NA, NA, 0,
#                                            0, NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(14) car sa p-value = 0.90165 > 0,05.
```

```
# reg<-Arima(rtt, order=c(0,0,23), fixed = c(NA, NA, 0, NA, 0, 0, 0, 0, NA, NA,
#                                            NA, NA, 0, 0, 0, 0, NA, NA, NA, 0,
#                                            0, NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(9) car sa p-value = 0.27212 > 0,05.
# reg<-Arima(rtt, order=c(0,0,23), fixed = c(NA, NA, 0, NA, 0, 0, 0, 0, 0, NA,
#                                            NA, NA, 0, 0, 0, 0, NA, NA, NA, 0,
#                                            0, NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(2) car sa p-value = 0.16902 > 0,05.
# reg<-Arima(rtt, order=c(0,0,23), fixed = c(NA, 0, 0, NA, 0, 0, 0, 0, 0, NA,
#                                            NA, NA, 0, 0, 0, 0, NA, NA, NA, 0,
#                                            0, NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(11) car sa p-value = 0.12096 > 0,05.
# reg<-Arima(rtt, order=c(0,0,23), fixed = c(NA, 0, 0, NA, 0, 0, 0, 0, 0, NA,
#                                            0, NA, 0, 0, 0, 0, NA, NA, NA, 0,
#                                            0, NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(17) car sa p-value = 0.11624  > 0,05.
# reg<-Arima(rtt, order=c(0,0,23), fixed = c(NA, 0, 0, NA, 0, 0, 0, 0, 0, NA,
#                                            0, NA, 0, 0, 0, 0, 0, NA, NA, 0,
#                                            0, NA, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(22) car sa p-value = 0.11600 > 0,05.
# reg<-Arima(rtt, order=c(0,0,23), fixed = c(NA, 0, 0, NA, 0, 0, 0, 0, 0, NA,
#                                            0, NA, 0, 0, 0, 0, 0, NA, NA, 0,
#                                            0, 0, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(19) car sa p-value = 0.08165 > 0,05.
# reg<-Arima(rtt, order=c(0,0,23), fixed = c(NA, 0, 0, NA, 0, 0, 0, 0, 0, NA,
#                                            0, NA, 0, 0, 0, 0, 0, NA, 0, 0,
#                                            0, 0, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(18) car sa p-value = 0.08373 > 0,05.
# reg<-Arima(rtt, order=c(0,0,23), fixed = c(NA, 0, 0, NA, 0, 0, 0, 0, 0, NA,
#                                            0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
#                                            0, 0, NA, NA))
# coeftest(reg)
#
# # Fixons à 0 theta(4) car sa p-value = 0.05128 > 0,05.
# reg<-Arima(rtt, order=c(0,0,23), fixed = c(NA, 0, 0, 0, 0, 0, 0, 0, 0, NA,
#                                            0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
#                                            0, 0, NA, NA))
# coeftest(reg)
#
```

```r
# # Fixons à 0 theta(10) car sa p-value = 0.05783 > 0,05.
# reg<-Arima(rtt, order=c(0,0,23), fixed = c(NA, 0, 0, 0, 0, 0, 0, 0, 0, 0,
#                                             0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
#                                             0, 0, NA, NA))
# coeftest(reg)
```

```r
# Enlevons la constante car elle n'est pas significatif (p-value = 0.777 > 0,05).
reg<-Arima(rtt, order=c(0,0,23),
           include.mean = F,
           fixed = c(NA, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                     0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
                     0, 0, NA))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value Pr(>|z|)
## ma1  -0.065388   0.027897 -2.3439  0.01908 *
## ma12 -0.064281   0.029803 -2.1569  0.03102 *
## ma23 -0.058606   0.030858 -1.8992  0.05754 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Fixons à 0 theta(23) car sa p-value = 0.05783 > 0,05 donc cela revient à
# faire un MA(12)
```

MA(12) En lien avec notre rho(12) qui sortait dans l'ACf des rendements logarithmique

```r
reg1 <-Arima(rtt, order=c(0,0,12),
           include.mean = F,
           fixed = c(NA, 0, 0, 0, 0, 0, 0, 0, 0, 0,0, NA))
coeftest(reg1)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value Pr(>|z|)
## ma1  -0.065670   0.027892 -2.3545  0.01855 *
## ma12 -0.067262   0.029692 -2.2653  0.02349 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Maintenant que nous avons un modele avec tous les coefficients qui sont significatif
# Celui-ci n'est valide que si les aléas sont des Bruits Blancs i.e.
# E[epsilon] = 0 et epsilon ne sont pas autocorrélés
```

```r
# Test de l'espérance nulle des erreurs
# H0 : E[epsilon] = 0 VS Ha : E[epsilon] =/= 0
residu<-reg1$res
t.test(residu)
```

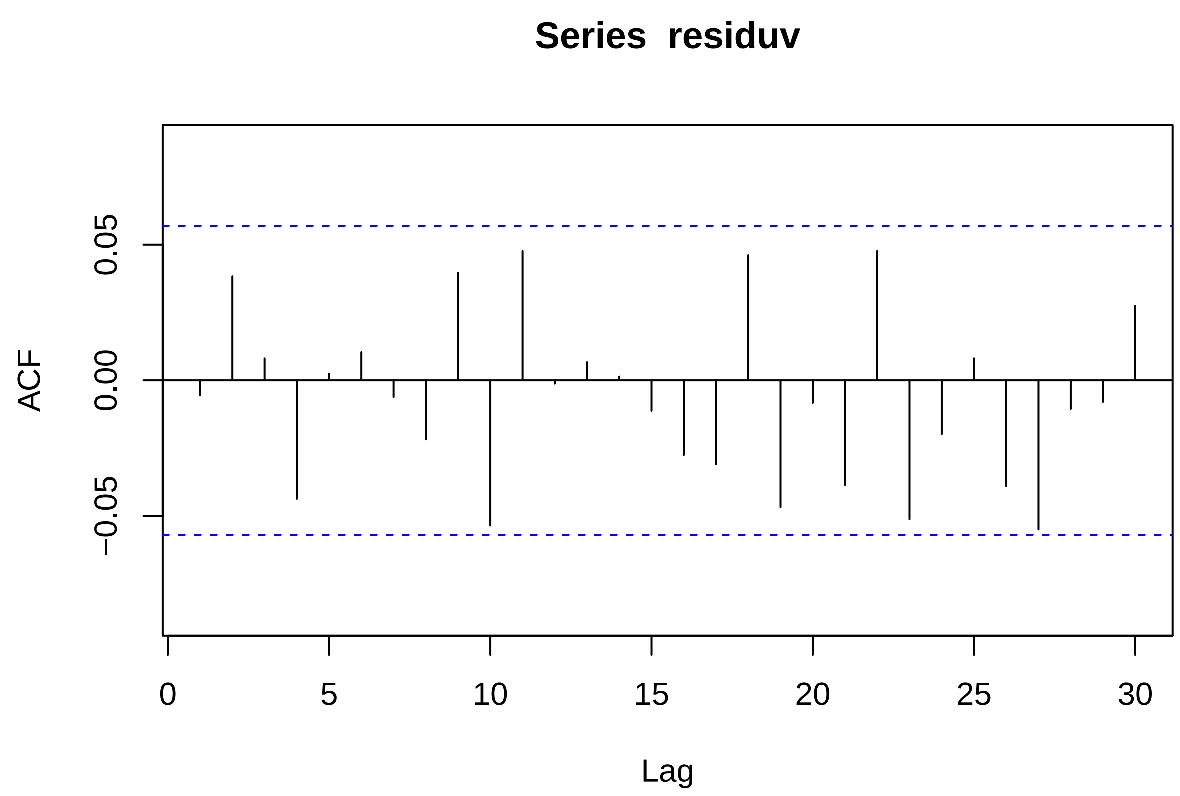
```
##
##  One Sample t-test
##
## data:  residu
## t = 1.6956, df = 1184, p-value = 0.09023
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0002009471  0.0027588691
## sample estimates:
##   mean of x
## 0.001278961
```

```
# p-value = 0.09023 > 0.05 donc on accepte H0
# L'espérance des aléas sont nulles
```

```
## Test d'abscence d'autocorrélation dans les aléas
library(tseries)
```

```
residuv=(residu-mean(residu))/sd(residu)
```

```
Acf(residuv) # On observe aucune autocorrélation
```




```
tmp<-rep(0,40)
for(i in 1:40){
```

```r
  tmp[i]<-Box.test(residuv,lag=i,type="Ljung-Box")$p.value
}
tmp
```

```
##  [1] 0.8497704 0.4102102 0.6020374 0.3883741 0.5294696 0.6404590 0.7430477
##  [8] 0.7703004 0.6619432 0.4239257 0.2991579 0.3753848 0.4503897 0.5289694
## [15] 0.5929295 0.5963746 0.5821871 0.4723205 0.3716608 0.4289201 0.3843482
## [22] 0.2963898 0.2087089 0.2330495 0.2748064 0.2438722 0.1587100 0.1882829
## [29] 0.2225410 0.2288917 0.2580022 0.2509386 0.2537636 0.2537723 0.2932672
## [36] 0.2864890 0.2993469 0.3400914 0.2706422 0.3086876
```

```r
# p-values > 5 %.
# On accepte H0, absence d'autocorrélation des aléas

# Notre modèle est donc valide.
```

AR(23) Pour la même raison que nous avons tester le MA(23), nous testons un AR(23)

```r
reg<-Arima(rtt, order=c(23,0,0))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ar1       -0.06041055  0.02902131 -2.0816  0.03738 *
## ar2        0.03185637  0.02906883  1.0959  0.27312
## ar3        0.01168340  0.02917817  0.4004  0.68885
## ar4       -0.05103599  0.02916471 -1.7499  0.08013 .
## ar5        0.00533232  0.02924302  0.1823  0.85531
## ar6        0.01197856  0.02919988  0.4102  0.68164
## ar7       -0.00869378  0.02917676 -0.2980  0.76573
## ar8       -0.01750876  0.02916431 -0.6003  0.54827
## ar9        0.03391424  0.02914575  1.1636  0.24458
## ar10      -0.04646688  0.02927104 -1.5875  0.11241
## ar11       0.03713160  0.02927410  1.2684  0.20465
## ar12      -0.05796481  0.02928357 -1.9794  0.04777 *
## ar13      -0.00693453  0.02930694 -0.2366  0.81295
## ar14       0.00734468  0.02928885  0.2508  0.80199
## ar15      -0.00583154  0.02932754 -0.1988  0.84239
## ar16      -0.03385186  0.02934529 -1.1536  0.24868
## ar17      -0.02995119  0.02937310 -1.0197  0.30788
## ar18       0.04607860  0.02940416  1.5671  0.11710
## ar19      -0.03840099  0.02944457 -1.3042  0.19217
## ar20      -0.02570491  0.02951595 -0.8709  0.38382
## ar21      -0.02924937  0.02956098 -0.9895  0.32244
## ar22       0.04904786  0.02957467  1.6584  0.09723 .
## ar23      -0.04820071  0.02983632 -1.6155  0.10620
## intercept  0.00109197  0.00061006  1.7899  0.07347 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# H0 : phi_k = 0 VS Ha : phi_k =/= 0
# Les coefficients ne sont pas tous significatifs.
```

```r
# Nous commence par fixer à 0 tous les phi ayant une p-value très élevée :
# phi(3), phi(5), phi(6), phi(7), phi(8), phi(13), phi(14), phi(15)
reg<-Arima(rtt, order=c(23,0,0), fixed = c(NA, NA, 0, NA, 0, 0, 0, 0, NA, NA,
                                           NA, NA, 0, 0, 0, NA, NA, NA, NA, NA,
                                           NA, NA, NA, NA))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error z value Pr(>|z|)
## ar1       -0.0597381  0.0289307 -2.0649  0.03894 *
## ar2        0.0295324  0.0289035  1.0218  0.30689
## ar4       -0.0521569  0.0288980 -1.8049  0.07110 .
## ar9        0.0349627  0.0290304  1.2044  0.22845
## ar10      -0.0480618  0.0292147 -1.6451  0.09994 .
## ar11       0.0373404  0.0292278  1.2776  0.20140
## ar12      -0.0561086  0.0291784 -1.9229  0.05449 .
## ar16      -0.0333521  0.0291594 -1.1438  0.25271
## ar17      -0.0297815  0.0291888 -1.0203  0.30758
## ar18       0.0452027  0.0292595  1.5449  0.12237
## ar19      -0.0390258  0.0293270 -1.3307  0.18328
## ar20      -0.0248832  0.0294562 -0.8448  0.39825
## ar21      -0.0287488  0.0295269 -0.9736  0.33023
## ar22       0.0476260  0.0295172  1.6135  0.10664
## ar23      -0.0475066  0.0296887 -1.6002  0.10956
## intercept  0.0010933  0.0006108  1.7899  0.07347 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# De même pour : phi(2), phi(9), phi(11), phi(16), phi(17), phi(20), phi(21)
reg<-Arima(rtt, order=c(23,0,0), fixed = c(NA, 0, 0, NA, 0, 0, 0, 0, 0, NA,
                                           0, NA, 0, 0, 0, 0, 0, NA, NA, 0,
                                           0, NA, NA, NA))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ar1       -0.06205472  0.02887507 -2.1491  0.03163 *
## ar4       -0.04744900  0.02888070 -1.6429  0.10040
## ar10      -0.05455324  0.02908665 -1.8755  0.06072 .
## ar12      -0.05980806  0.02911949 -2.0539  0.03999 *
## ar18       0.04560525  0.02918088  1.5628  0.11809
## ar19      -0.04364825  0.02923024 -1.4933  0.13537
## ar22       0.04923666  0.02947338  1.6705  0.09481 .
## ar23      -0.05327835  0.02968476 -1.7948  0.07268 .
## intercept  0.00109974  0.00061181  1.7975  0.07225 .
```

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

```r
# Fixons à 0 phi(19) car sa p-value = 0.13537 > 0,05.
reg<-Arima(rtt, order=c(23,0,0), fixed = c(NA, 0, 0, NA, 0, 0, 0, 0, 0, NA,
                                           0, NA, 0, 0, 0, 0, 0, NA, 0, 0,
                                           0, NA, NA, NA))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ar1       -0.06408057  0.02887072 -2.2196  0.02645 *
## ar4       -0.04696957  0.02890518 -1.6250  0.10417
## ar10      -0.05645791  0.02908753 -1.9410  0.05226 .
## ar12      -0.06002823  0.02914737 -2.0595  0.03945 *
## ar18       0.04911379  0.02911377  1.6870  0.09161 .
## ar22       0.04904349  0.02950172  1.6624  0.09643 .
## ar23      -0.05162511  0.02969139 -1.7387  0.08208 .
## intercept  0.00110210  0.00063526  1.7349  0.08276 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Fixons à 0 phi(4) car sa p-value = 0.10417 > 0,05.
reg<-Arima(rtt, order=c(23,0,0), fixed = c(NA, 0, 0, 0, 0, 0, 0, 0, 0, NA,
                                           0, NA, 0, 0, 0, 0, 0, NA, 0, 0,
                                           0, NA, NA, NA))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ar1       -0.06464214  0.02890116 -2.2367  0.02531 *
## ar10      -0.05693082  0.02911852 -1.9551  0.05057 .
## ar12      -0.05914027  0.02917514 -2.0271  0.04265 *
## ar18       0.04908863  0.02914559  1.6843  0.09213 .
## ar22       0.04694463  0.02950568  1.5910  0.11160
## ar23      -0.04921971  0.02968815 -1.6579  0.09734 .
## intercept  0.00109916  0.00066225  1.6597  0.09697 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Fixons à 0 phi(22) car sa p-value = 0.11160 > 0,05.
reg<-Arima(rtt, order=c(23,0,0), fixed = c(NA, 0, 0, 0, 0, 0, 0, 0, 0, NA,
                                           0, NA, 0, 0, 0, 0, 0, NA, 0, 0,
                                           0, 0, NA, NA))
coeftest(reg)
```

```
## Warning in sqrt(diag(se)): Production de NaN
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value  Pr(>|z|)
## ar1       -0.06663068         NaN     NaN       NaN
## ar10      -0.06142674  0.01738008 -3.5343 0.0004088 ***
## ar12      -0.06080444         NaN     NaN       NaN
## ar18       0.04894644         NaN     NaN       NaN
## ar23      -0.05222402         NaN     NaN       NaN
## intercept  0.00110246  0.00063112  1.7468 0.0806652 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Fixons à 0 phi(18) car sa p-value = 0.11054 > 0,05.
reg<-Arima(rtt, order=c(23,0,0), fixed = c(NA, 0, 0, 0, 0, 0, 0, 0, 0, NA,
                                           0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
                                           0, 0, NA, NA))
coeftest(reg)
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ar1       -0.06765538  0.02893495 -2.3382  0.01938 *
## ar10      -0.06109686  0.02910863 -2.0989  0.03582 *
## ar12      -0.06144019  0.02918157 -2.1054  0.03525 *
## ar23      -0.05295274  0.02963358 -1.7869  0.07395 .
## intercept  0.00109235  0.00060633  1.8016  0.07161 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Fixons à 0 theta(23) car sa p-value = 0.08396 > 0,05 donc cela revient à
# faire un AR(12)
```

```r
reg<-Arima(rtt, order=c(12,0,0), fixed = c(NA, 0, 0, 0, 0, 0, 0, 0, 0, NA,
                                           0, NA, NA))
coeftest(reg)
```

AR(12)

```
##
## z test of coefficients:
##
##              Estimate  Std. Error z value Pr(>|z|)
## ar1       -0.07039397  0.02483485 -2.8345  0.00459 **
## ar10      -0.06043203  0.02805468 -2.1541  0.03123 *
## ar12      -0.06360131  0.02737294 -2.3235  0.02015 *
## intercept  0.00110089  0.00063137  1.7437  0.08122 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# le modèle ne converge pas, il n'est donc pas valide.
```

On choisi donc notre MA(12)

PROPRIETE 4 : Clusters de volatilité

```r
# Nous analysons maintenant l'existence de clusters de volatilité.
# Ce phénomène, remet en cause l'hypothèse d'homoscédasticité conditionnelle.
# Afin de vérifier formellement si la variance dépend des chocs passés, nous
# appliquons le test ARCH d'Engle (1982) sur la série  rtt.
library(FinTS)
```

```r
LM1<-ArchTest(as.numeric(rtt),lag=1)
LM1 #  p-value = 6.813e-05 < 5%
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  as.numeric(rtt)
## Chi-squared = 15.862, df = 1, p-value = 6.813e-05
```

```r
# On rejette l'hypothèse nulle d'homoscédasticité conditionnelle donc présence de
# clusters de volatilités
```

PROPRIETE 5 : Queues épaisses conditionnelles

```r
library(tseries)
library(fGarch)
library(FinTS)
library(moments)

# Après avoir estimé l'équation de la moyenne avec un modèle MA(12) et vérifié que
# ses résidus ne présentent plus d'autocorrélation, le test d'Engle met en évidence
# la présence de volatilité conditionnelle. Il est donc nécessaire de modéliser la
# variance avec un modèle GARCH, en particulier le GARCH(1,1).
```

```r
volat<-garch(residuv,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  5.841e+02
##      1     4  5.831e+02  1.77e-03  1.14e-02  5.5e-02  6.8e+02  9.9e-02  3.86e+00
##      2     6  5.819e+02  2.06e-03  2.86e-03  1.6e-02  2.9e+00  3.9e-02  1.38e+00
##      3     8  5.808e+02  1.82e-03  2.94e-03  6.4e-02  4.5e+00  1.3e-01  2.24e-01
##      4     9  5.787e+02  3.61e-03  3.87e-03  7.8e-02  2.0e+00  1.3e-01  5.53e-01
##      5    10  5.774e+02  2.33e-03  3.92e-03  7.0e-02  2.0e+00  1.3e-01  1.06e+00
##      6    11  5.758e+02  2.71e-03  3.81e-03  1.0e-01  2.0e+00  1.3e-01  4.09e-01
##      7    12  5.744e+02  2.39e-03  2.90e-03  1.1e-01  1.9e+00  1.3e-01  9.97e-02
##      8    14  5.677e+02  1.17e-02  1.46e-02  3.1e-01  1.5e+00  5.1e-01  7.60e-02
##      9    17  5.670e+02  1.27e-03  3.08e-03  5.6e-03  9.9e+00  1.0e-02  4.35e-01
##     10    18  5.661e+02  1.62e-03  1.78e-03  5.8e-03  2.0e+00  1.0e-02  3.79e-01
##     11    19  5.651e+02  1.69e-03  1.78e-03  1.1e-02  2.0e+00  2.1e-02  4.13e-01
##     12    20  5.641e+02  1.82e-03  2.66e-03  1.7e-02  2.0e+00  4.2e-02  4.37e-01
##     13    21  5.635e+02  1.09e-03  2.39e-03  2.1e-02  2.0e+00  4.2e-02  9.91e-02
##     14    23  5.633e+02  3.82e-04  1.24e-03  5.5e-03  2.0e+00  1.5e-02  1.56e-02
##     15    24  5.630e+02  5.06e-04  7.52e-04  6.2e-03  1.9e+00  1.5e-02  9.63e-03
##     16    26  5.629e+02  1.74e-04  3.18e-04  3.5e-03  1.6e+00  7.3e-03  1.87e-03
##     17    27  5.629e+02  2.61e-05  4.41e-05  3.3e-03  1.9e+00  7.3e-03  8.96e-04
##     18    30  5.629e+02  1.91e-06  4.01e-06  6.6e-05  2.2e+01  1.4e-04  4.60e-05
##     19    31  5.629e+02  8.63e-07  1.04e-06  5.7e-05  5.2e+00  1.4e-04  9.65e-06
##     20    32  5.629e+02  7.32e-07  1.21e-06  1.4e-04  1.9e+00  2.8e-04  8.64e-06
##     21    33  5.629e+02  1.28e-06  1.86e-06  2.7e-04  1.5e+00  5.6e-04  7.55e-06
##     22    34  5.629e+02  1.35e-06  2.56e-06  5.2e-04  9.8e-01  1.1e-03  5.61e-06
##     23    35  5.629e+02  5.10e-07  2.85e-06  1.0e-03  4.0e-01  2.3e-03  3.13e-06
##     24    40  5.629e+02  3.83e-10  1.16e-08  4.2e-06  5.1e+00  1.1e-05  3.31e-07
##     25    54  5.629e+02  2.02e-16  2.21e-15  1.0e-12  2.4e+07  2.0e-12  3.21e-07
##     26    57  5.629e+02  0.00e+00  1.96e-17  9.1e-15  2.8e+09  1.8e-14  3.24e-07
##
##  ***** FALSE CONVERGENCE *****
##
##  FUNCTION     5.628607e+02   RELDX        9.142e-15
##  FUNC. EVALS      57         GRAD. EVALS      26
##  PRELDF       1.959e-17      NPRELDF      3.237e-07
##
##      I      FINAL X(I)        D(I)          G(I)
##
##      1    6.026386e-02     1.000e+00    -5.696e-01
##      2    6.637024e-02     1.000e+00     2.298e-01
##      3    8.743644e-01     1.000e+00     1.257e-01
```

```
# Ne converge pas donc faire garchFit
```

```
volat<-garchFit(~garch(1,1),data=residuv,trace=F,include.mean = FALSE)
summary(volat)
```

```
##
## Title:
```

```
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = residuv, include.mean = FALSE,
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x0000028fd535f4d8>
##  [data = residuv]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##     omega    alpha1     beta1
## 0.059846  0.066157  0.875017
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega    0.05985     0.02964    2.019  0.04344 *
## alpha1   0.06616     0.02234    2.961  0.00307 **
## beta1    0.87502     0.04724   18.523  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  -1652.537    normalized:  -1.394546
##
## Description:
##  Sat Dec  6 14:49:41 2025 by user: celes
##
##
## Standardised Residuals Tests:
##                                   Statistic      p-Value
##  Jarque-Bera Test   R    Chi^2  259.8420534 0.000000e+00
##  Shapiro-Wilk Test  R    W        0.9727553 3.809483e-14
##  Ljung-Box Test     R    Q(10)    6.4013540 7.804919e-01
##  Ljung-Box Test     R    Q(15)    8.8341612 8.860277e-01
##  Ljung-Box Test     R    Q(20)   13.5813440 8.510729e-01
##  Ljung-Box Test     R^2  Q(10)    5.7480874 8.359651e-01
##  Ljung-Box Test     R^2  Q(15)    9.9637069 8.220166e-01
##  Ljung-Box Test     R^2  Q(20)   16.9067400 6.590209e-01
##  LM Arch Test       R    TR^2     8.4461814 7.493636e-01
##
## Information Criterion Statistics:
##      AIC      BIC      SIC     HQIC
## 2.794155 2.807009 2.794142 2.799000
```

```
# H0 : alpha(k) = 0 VS Ha : alpha(k) =/= 0
# On a nos coefficients qui sont significatifs (p-value < 5%).
```

```r
# On souhaite maintenant savoir si avec notre modèle on a réussi à prendre en compte
# toutes l'hétéroscédasticité conditionnelle

resvolat=volat@residuals/volat@sigma.t
ArchTest(resvolat, lag = 1)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  resvolat
## Chi-squared = 0.015375, df = 1, p-value = 0.9013
```

```r
# p-value 0.9013 > 5% donc pas de cluster de volatilité
```

```r
ArchTest(resvolat, lag = 20)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  resvolat
## Chi-squared = 17.026, df = 20, p-value = 0.6513
```

```r
# p-value 0.6513 > 5% donc pas de cluster de volatilité jusqu'à l'ordre 20

# Les tests ARCH appliqués aux résidus du modèle GARCH(1,1) donnent des p-values
# supérieures à 5 % (pour les lags 1 et 20), ce qui conduit à accepter
# l’hypothèse d’absence d’effets ARCH.
```

```r
anscombe.test(resvolat)
```

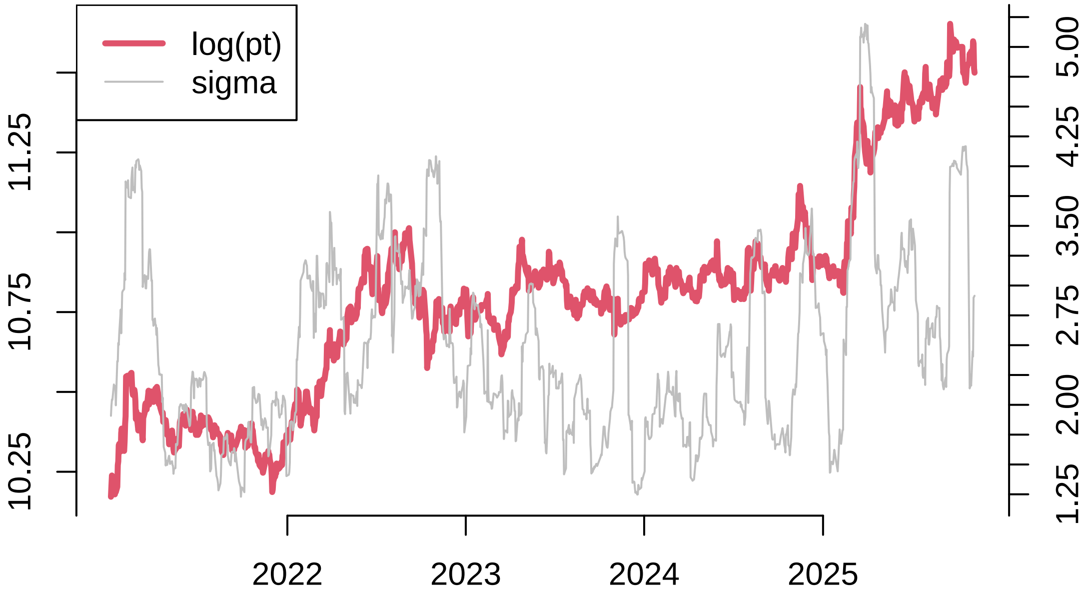
```
##
##  Anscombe-Glynn kurtosis test
##
## data:  resvolat
## kurt = 5.2520, z = 8.1434, p-value = 3.844e-16
## alternative hypothesis: kurtosis is not equal to 3
```

```r
# p-value = 3.845e-16 < 5 %, nous rejetons l’hypothèse nulle selon laquelle la kurtosis
# serait égale à 3.
# Ainsi, nous pouvons interpréter sa valeur calculée : kurt = 5.2520 >> 3, cela
# indique une distribution fortement leptokurtique
```

```r
# Même après la prise en compte de la volatilité conditionnelle, les résidus
# présentent donc encore des queues épaisses, confirmant la présence de leptokurtose conditionnelle.
```

PROPRIETE 6 : Effet de levier

```r
T <- length(rt)

sigma_full <- rep(NA, T)
for (t in 22:T) {
  sigma_full[t] <- sd(rt[(t-21):t])
}

idx_test <- which(dates %in% dates_rtt)

dates_sub <- dates[idx_test]
pt_sub    <- pt[idx_test]
sigma_sub <- sigma_full[idx_test] * 100

par(mar = c(5, 4, 4, 4) + 0.1)

plot(dates_sub, log(pt_sub),
     type = "l", col = 2, axes = FALSE, xlab = "", ylab = "", lwd = 3)

axis(2, at = seq(9.5, 11.5, by = 0.25))

years <- unique(format(dates_sub, "%Y"))
ticks <- as.Date(paste0(years, "-01-01"))
ticks <- ticks[ticks >= min(dates_sub) & ticks <= max(dates_sub)]
axis(1, at = ticks, labels = format(ticks, "%Y"))

par(new = TRUE)

plot(dates_sub, sigma_sub,
     type = "l", col = "grey", axes = FALSE, xlab = "", ylab = "")

axis(4, at = seq(0, 7.5, by = 0.25))

legend("topleft",
       legend = c("log(pt)", "sigma"),
       col = c(2, "grey"),
       lty = c(1, 1),
       lwd = c(3, 1))
```




```
# semble y avoir effet de levier en fin 2022
```

```
# Sigma + plus forte hausse / plus forte baisse
dpt_test <- dpt[idx_test]
idx_valid <- idx_test[!is.na(sigma_full[idx_test])]

sigma_valid_pct <- sigma_full[idx_valid] * 100
dpt_valid       <- dpt[idx_valid]

i_min <- which.min(dpt_valid)
i_max <- which.max(dpt_valid)

date_min_dpt <- dates[idx_valid][i_min]
date_max_dpt <- dates[idx_valid][i_max]

sigma_at_min_dpt <- sigma_valid_pct[i_min]
sigma_at_max_dpt <- sigma_valid_pct[i_max]

res_test <- data.frame(Type = c("Plus forte baisse", "Plus forte hausse"),
                       Date = c(date_min_dpt, date_max_dpt),
                       Rendement = c(dpt_valid[i_min], dpt_valid[i_max]),
                       Sigma = c(sigma_at_min_dpt, sigma_at_max_dpt))

print(res_test, row.names = FALSE)
```

```
##                 Type       Date Rendement     Sigma
```

```
##  Plus forte baisse 2025-03-18 -9933.062 5.085917
##  Plus forte hausse 2025-09-17 14800.000 3.773315
```

```
# sigma(Plus forte baisse) > sigma(Plus forte hausse) = effet de levier
# Cette chute intervient juste après l'annonce qu'un hélicoptère Surion de KAI a
# été totalement détruit lors d'une collision avec un drone Heron, ce qui a pu
# alimenter des craintes sur les risques opérationnels et provoquer une correction
# du titre.
# Cette forte hausse correspond au jour où KAI dévoile le design de son nouvel
# avion de guerre électronique lors du 2025 Electromagnetic Warfare Workshop à
# Séoul, annonce perçue très positivement par le marché
```

PROPRIETE 7 : La saisonnalité

```
library(moments)
```

Journalier - Effet week-end

```
# Dans la littérature, plusieurs travaux montrent que l'accumulation d'informations
# pendant les périodes de fermeture augmente la volatilité, en particulier en
# début de semaine (French & Roll, 1986) ou à partir du mercredi (Baillie & Bollerslev, 1989).
```

```
# On crée un vecteur de dates aligné avec rtt
dates_rtt <- tail(dates, length(rtt))
jour <- format(dates_rtt, format = "%A")

tableaures <- data.frame(matrix(NA,ncol=5,nrow=4))
colnames(tableaures) <- c("lundi","mardi","mercredi","jeudi","vendredi")
rownames(tableaures) <- c("moyenne en %","écart-type annuel en %","skewness","kurtosis")

rtmar <- as.numeric(rtt[jour=="mardi"])
mardi <- mean(rtmar)
tableaures[1,2] <- mardi*100
tableaures[2,2] <- sd(rtmar)*100*sqrt(252)
tableaures[3,2] <- skewness(rtmar)
tableaures[4,2] <- kurtosis(rtmar)

rtmer <- as.numeric(rtt[jour=="mercredi"])
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)

rtjeu <- as.numeric(rtt[jour=="jeudi"])
jeudi <- mean(rtjeu)
tableaures[1,4] <- jeudi*100
```

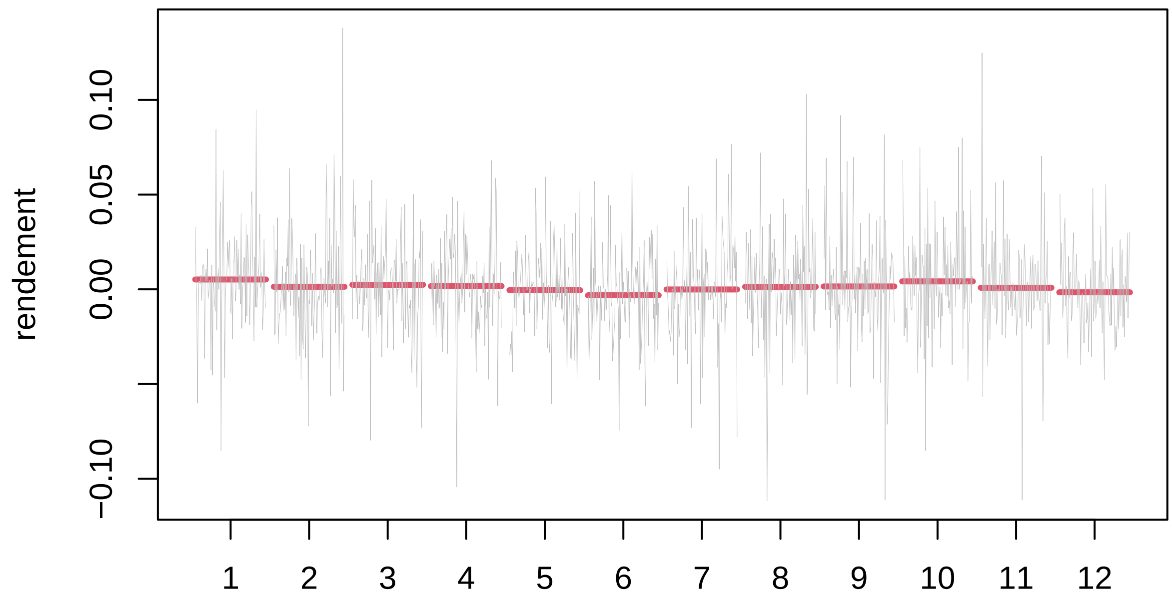
```r
tableaures[2,4] <- sd(rtjeu)*100*sqrt(252)
tableaures[3,4] <- skewness(rtjeu)
tableaures[4,4] <- kurtosis(rtjeu)

rtven <- as.numeric(rtt[jour=="vendredi"])
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)

rtlun <- as.numeric(rtt[jour=="lundi"])
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)

tableaures
```

```
##                              lundi      mardi    mercredi      jeudi    vendredi
## moyenne en %             0.3110431  0.1937490 -0.08829934  0.1173382  0.02949074
## écart-type annuel en % 44.0735909 43.8655173 43.08766347 38.8451832 37.09392310
## skewness               -0.2598035 -0.4626869  0.58057093  0.6564189  0.16922622
## kurtosis                2.2340606  2.5496950  3.09429899  4.9498742  1.10482656
```

```r
# On observe bien un effet week-end
# En effet, lundi est le jour avec la moyenne la plus importante ainsi que
# l'écart-type annuel le plus important.
```

Mensuelle - Effet janvier

```r
monthplot(rtt, ylab="rendement", main="", cex.main=1,
          col.base= 2, lwd.base=3,
          col='grey', ,lwd=0.01)
```




```
# Nous observons un effet janvier dans nos données.
# Les rendements moyens les plus élevés apparaissent en janvier suivi d'octobre,
# tandis que les mois les plus faibles paraissent être en juin et décembre.

# La volatilité la plus importante est en février.
```

PROPRIETE 8 : Stationnarité

Test de Dickey-Fuller (DF)

```
library(urca)
```

```
summary(ur.df(rtt,type= "trend",lags=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
## -0.109502 -0.013573 -0.000385  0.013057  0.138381
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.144e-04  1.515e-03   0.604    0.546
## z.lag.1     -1.077e+00  2.900e-02 -37.131   <2e-16 ***
## tt           4.237e-07  2.214e-06   0.191    0.848
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02604 on 1181 degrees of freedom
## Multiple R-squared:  0.5386, Adjusted R-squared:  0.5378
## F-statistic: 689.4 on 2 and 1181 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -37.131 459.5705 689.3542
##
## 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
```

```
# H0 : B1 = 0 vs Ha : B1 =/= 0
# p-value(B1) = 0.848 > 0.05 le coefficient n'est pas significatif
# Donc on estime drift
```

```
summary(ur.df(rtt,type= "drift",lags=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
## -0.109460 -0.013613 -0.000383  0.013064  0.138618
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  0.0011654  0.0007571   1.539    0.124
## z.lag.1     -1.0767940  0.0289885 -37.146   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02603 on 1182 degrees of freedom
## Multiple R-squared:  0.5386, Adjusted R-squared:  0.5382
## F-statistic:  1380 on 1 and 1182 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -37.1456 689.8998
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

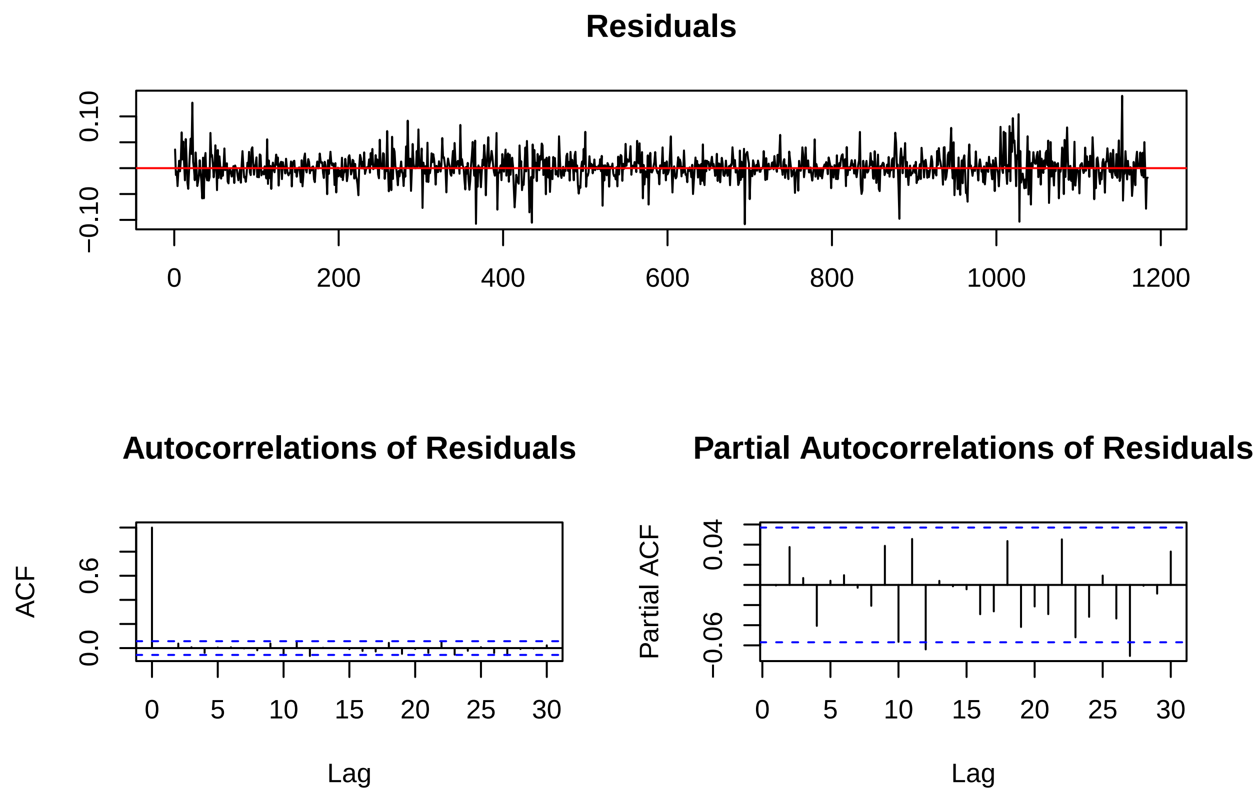
```
# H0 : B0 = 0 vs Ha : B0 =/= 0
# p-value(B0) = 0.124 > 0.05 le coefficient n'est pas significatif
# Donc on estime none
```

```
summary(ur.df(rtt,type= "none",lags=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
## -0.108368 -0.012475  0.000762  0.014245  0.139733
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## z.lag.1 -1.07488    0.02898  -37.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02604 on 1183 degrees of freedom
## Multiple R-squared:  0.5377, Adjusted R-squared:  0.5373
## F-statistic:  1376 on 1 and 1183 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -37.0923
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```
# H0 : rho-1 = 0 vs Ha : rho-1 =/= 0
# On a t.calculé = -37.0923 < -1.95 donc on rejette H0

# Ainsi on peut conclure à partir du test de DF que le PGD est stationnaire.
# valide que si les aléas de la régression de Dickey et Fuller ne sont pas
# auto-corrélés.
```

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




```
# La figure nous indique que les aléas sont auto-corrélés aux ordres 12 et 27 et donc
# notre conclusion concernant l'absence de RU n'est pas valide.
# Nous devons effectuer un test de RU dans le cadre de la régression de Dickey Fuller Augmenté.
```

Test de Dickey Fuller Augmenté (ADF)

```
library(CADFtest)
```

```
T = length(rtt)
Schwert<-as.integer(12*(T/100)^(0.25)) # = 22
```

```r
# Nous vérifions que B0 est bien non significatif
summary(CADFtest(rtt, criterion="MAIC",type="drift",max.lag.y=Schwert))
```

```
## Augmented DF test
##                                                ADF test
## t-test statistic:                          -1.991076e+01
## p-value:                                    1.247208e-42
## Max lag of the diff. dependent variable:    2.000000e+00
##
## Call:
## dynlm(formula = formula(model), start = obs.1, end = obs.T)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.108216 -0.013512 -0.000188  0.012929  0.139976
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0008460  0.0007544   1.121    0.262
## L(y, 1)     -1.0480253  0.0526361 -19.911   <2e-16 ***
## L(d(y), 1)  -0.0398867  0.0429322  -0.929    0.353
## L(d(y), 2)  -0.0101856  0.0291541  -0.349    0.727
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02565 on 1158 degrees of freedom
## Multiple R-squared:  0.5503, Adjusted R-squared:  0.5491
## F-statistic: 0.5534 on 2 and 1158 DF,  p-value: 0.5751
```

```r
# Le MAIC nous donne un lag de 2
```

```r
summary(ur.df(rte,type= "drift",lags=2))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35238 -0.01209  0.00015  0.01140  0.16414
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0001595  0.0005276  -0.302    0.762
## z.lag.1     -1.0338469  0.0374139 -27.633   <2e-16 ***
```

```
## z.diff.lag1  0.0177146  0.0303541   0.584    0.560
## z.diff.lag2  0.0061935  0.0212947   0.291    0.771
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0248 on 2206 degrees of freedom
## Multiple R-squared:  0.5079, Adjusted R-squared:  0.5073
## F-statistic:   759 on 3 and 2206 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -27.6327 381.7838
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

```
# p-value(B0) = 0.762 > 0.05 B0 n'est pas significatif
```

```
# Donc on estime bien none
summary(CADFtest(rtt, criterion="MAIC",type="none",max.lag.y=Schwert))
```

```
## Augmented DF test
##                                                ADF test
## t-test statistic:                          -1.135396e+01
## p-value:                                    6.865332e-23
## Max lag of the diff. dependent variable:    8.000000e+00
##
## Call:
## dynlm(formula = formula(model), start = obs.1, end = obs.T)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.107990 -0.013022  0.000685  0.013424  0.142726
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## L(y, 1)    -1.038612   0.091476 -11.354   <2e-16 ***
## L(d(y), 1) -0.046150   0.086066  -0.536    0.592
## L(d(y), 2) -0.013165   0.080537  -0.163    0.870
## L(d(y), 3) -0.005279   0.074553  -0.071    0.944
## L(d(y), 4) -0.041317   0.067854  -0.609    0.543
## L(d(y), 5) -0.033485   0.060368  -0.555    0.579
## L(d(y), 6) -0.016733   0.052402  -0.319    0.750
## L(d(y), 7) -0.014571   0.042803  -0.340    0.734
## L(d(y), 8) -0.043139   0.029113  -1.482    0.139
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02567 on 1153 degrees of freedom
## Multiple R-squared:  0.5518, Adjusted R-squared:  0.5483
## F-statistic: 0.797 on 8 and 1153 DF,  p-value: 0.6053
```

```r
# Le MAIC nous donne un lag de 8
```

```r
# Nous vérifions que B0 est bien non significatif
summary(ur.df(rtt,type= "drift",lags=8))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.10920 -0.01418 -0.00039  0.01273  0.14203
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0011232  0.0007678   1.463    0.144
## z.lag.1     -1.0315956  0.0925070 -11.152   <2e-16 ***
## z.diff.lag1 -0.0424507  0.0869627  -0.488    0.626
## z.diff.lag2 -0.0025530  0.0812572  -0.031    0.975
## z.diff.lag3  0.0079203  0.0751488   0.105    0.916
## z.diff.lag4 -0.0327352  0.0683680  -0.479    0.632
## z.diff.lag5 -0.0303190  0.0607937  -0.499    0.618
## z.diff.lag6 -0.0190770  0.0527925  -0.361    0.718
## z.diff.lag7 -0.0235337  0.0431339  -0.546    0.585
## z.diff.lag8 -0.0447155  0.0293808  -1.522    0.128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02608 on 1166 degrees of freedom
## Multiple R-squared:  0.5416, Adjusted R-squared:  0.5381
## F-statistic: 153.1 on 9 and 1166 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -11.1515 62.1818
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

```r
# p-value(B0) = 0.144 > 0.05 B0 n'est pas significatif
# Donc on estime none
```

```r
summary(ur.df(rtt,type= "none",lags=8))
```

```
##
```

```
## ###############################################
## # 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
## -0.108008 -0.013113  0.000709  0.013984  0.143073
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -1.012861   0.091661 -11.050   <2e-16 ***
## z.diff.lag1 -0.059218   0.086246  -0.687    0.492
## z.diff.lag2 -0.017231   0.080675  -0.214    0.831
## z.diff.lag3 -0.004604   0.074696  -0.062    0.951
## z.diff.lag4 -0.043094   0.068034  -0.633    0.527
## z.diff.lag5 -0.038563   0.060562  -0.637    0.524
## z.diff.lag6 -0.025283   0.052648  -0.480    0.631
## z.diff.lag7 -0.027761   0.043058  -0.645    0.519
## z.diff.lag8 -0.046806   0.029360  -1.594    0.111
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02609 on 1167 degrees of freedom
## Multiple R-squared:  0.5408, Adjusted R-squared:  0.5372
## F-statistic: 152.7 on 9 and 1167 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -11.0501
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```r
# On valide la spécification lorsque le dernier gamma est significatif.
# t.stat = |-1.594| < 1.64 donc dernier gamma non significatif
```

```r
summary(ur.df(rtt,type= "none",lags=7))
```

```
##
## ###############################################
## # 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
## -0.10830 -0.01270  0.00071  0.01413  0.14269
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -1.062232   0.086231 -12.318   <2e-16 ***
## z.diff.lag1 -0.010982   0.080690  -0.136    0.892
## z.diff.lag2  0.031247   0.074710   0.418    0.676
## z.diff.lag3  0.044421   0.068032   0.653    0.514
## z.diff.lag4  0.005398   0.060583   0.089    0.929
## z.diff.lag5  0.007984   0.052666   0.152    0.880
## z.diff.lag6  0.021568   0.043064   0.501    0.617
## z.diff.lag7  0.021774   0.029363   0.742    0.459
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0261 on 1169 degrees of freedom
## Multiple R-squared:  0.5398, Adjusted R-squared:  0.5367
## F-statistic: 171.4 on 8 and 1169 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -12.3185
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```
# t.stat = |0.742| < 1.64 donc dernier gamma non significatif
```

```
summary(ur.df(rtt,type= "none",lags=6))
```

```
##
## ###############################################
## # 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
## -0.107533 -0.012769  0.000765  0.014170  0.142887
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -1.039809   0.080611 -12.899   <2e-16 ***
## z.diff.lag1 -0.033550   0.074649  -0.449    0.653
```

```
## z.diff.lag2  0.008423   0.067976   0.124    0.901
## z.diff.lag3  0.021821   0.060520   0.361    0.718
## z.diff.lag4 -0.016313   0.052633  -0.310    0.757
## z.diff.lag5 -0.013882   0.043034  -0.323    0.747
## z.diff.lag6 -0.001503   0.029343  -0.051    0.959
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02608 on 1171 degrees of freedom
## Multiple R-squared:  0.5397, Adjusted R-squared:  0.5369
## F-statistic: 196.1 on 7 and 1171 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -12.8991
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```
# t.stat = |-0.051| < 1.64 donc dernier gamma non significatif
```

```
summary(ur.df(rtt,type= "none",lags=5))
```

```
##
## ###############################################
## # 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
## -0.107578 -0.012725  0.000756  0.014235  0.142925
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -1.041221   0.074571 -13.963   <2e-16 ***
## z.diff.lag1 -0.032203   0.067916  -0.474    0.635
## z.diff.lag2  0.009152   0.060449   0.151    0.880
## z.diff.lag3  0.022294   0.052551   0.424    0.671
## z.diff.lag4 -0.016082   0.043000  -0.374    0.708
## z.diff.lag5 -0.012921   0.029309  -0.441    0.659
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02606 on 1173 degrees of freedom
## Multiple R-squared:  0.5397, Adjusted R-squared:  0.5373
## F-statistic: 229.2 on 6 and 1173 DF,  p-value: < 2.2e-16
##
```

```
##
## Value of test-statistic is: -13.9629
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```r
# t.stat = |-0.441| < 1.64 donc dernier gamma non significatif
```

```r
summary(ur.df(rtt,type= "none",lags=4))
```

```
##
## ###############################################
## # 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
## -0.107156 -0.012758  0.000821  0.014172  0.142652
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -1.054929   0.067805 -15.558   <2e-16 ***
## z.diff.lag1 -0.018282   0.060377  -0.303    0.762
## z.diff.lag2  0.022634   0.052498   0.431    0.666
## z.diff.lag3  0.036000   0.042937   0.838    0.402
## z.diff.lag4 -0.002016   0.029282  -0.069    0.945
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02604 on 1175 degrees of freedom
## Multiple R-squared:  0.5398, Adjusted R-squared:  0.5379
## F-statistic: 275.7 on 5 and 1175 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -15.5584
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```r
# t.stat = |-0.069| < 1.64 donc dernier gamma non significatif
```

```r
summary(ur.df(rtt,type= "none",lags=3))
```

```
##
```

```
## ###############################################
## # 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
## -0.10721 -0.01286  0.00072  0.01410  0.14274
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -1.05908    0.06032 -17.559   <2e-16 ***
## z.diff.lag1 -0.01387    0.05247  -0.264    0.792
## z.diff.lag2  0.02770    0.04290   0.646    0.519
## z.diff.lag3  0.03968    0.02920   1.359    0.174
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02604 on 1177 degrees of freedom
## Multiple R-squared:  0.5394, Adjusted R-squared:  0.5378
## F-statistic: 344.6 on 4 and 1177 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -17.559
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```r
# t.stat = |1.359| < 1.64 donc dernier gamma non significatif
```

```r
summary(ur.df(rtt,type= "none",lags=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
## -0.10774 -0.01265  0.00078  0.01436  0.14072
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -1.01903    0.05234 -19.468   <2e-16 ***
## z.diff.lag1 -0.05433    0.04276  -1.271    0.204
## z.diff.lag2 -0.01460    0.02919  -0.500    0.617
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02604 on 1179 degrees of freedom
## Multiple R-squared:  0.5387, Adjusted R-squared:  0.5375
## F-statistic: 458.9 on 3 and 1179 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -19.4678
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```
# t.stat = |-0.500| < 1.64 donc dernier gamma non significatif
```

```
summary(ur.df(rtt,type= "none",lags=1))
```

```
##
## ###############################################
## # 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
## -0.108023 -0.012574  0.000667  0.014244  0.140276
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -1.03548    0.04260 -24.305   <2e-16 ***
## z.diff.lag -0.03801    0.02904  -1.309    0.191
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02602 on 1181 degrees of freedom
## Multiple R-squared:  0.5391, Adjusted R-squared:  0.5383
## F-statistic: 690.7 on 2 and 1181 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -24.3053
##
## Critical values for test statistics:
```

```
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```r
# t.stat = |-1.309| < 1.64 donc dernier gamma non significatif
```

```r
# Donc cela revient à DF : le PGD est stationnaire.
```

Test de Zivot et Andrews (ZA)

```r
# Cependant, ces résultat non sont valide que si pendant toute la période d'étude,
# il n'y a pas de choc structurelle u conjoncturelle
# HO : DS sans date de rupture
# Ha : TS avec une date de rupture
```

```r
# On commence avec comme choix du model "both"
summary(ur.za(rtt, model = "both", lag = Schwert))
```

```
##
## ################################
## # Zivot-Andrews Unit Root Test #
## ################################
##
##
## Call:
## lm(formula = testmat)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.113017 -0.012956 -0.000284  0.013085  0.140745
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.026e-03  1.696e-03   0.605 0.545373
## y.l1        -4.658e-01  1.609e-01  -2.896 0.003855 **
## trend       -9.370e-07  2.872e-06  -0.326 0.744277
## y.dl1        3.802e-01  1.563e-01   2.433 0.015126 *
## y.dl2        3.943e-01  1.512e-01   2.608 0.009237 **
## y.dl3        3.946e-01  1.464e-01   2.695 0.007139 **
## y.dl4        3.412e-01  1.418e-01   2.407 0.016262 *
## y.dl5        3.412e-01  1.373e-01   2.486 0.013055 *
## y.dl6        3.480e-01  1.322e-01   2.633 0.008575 **
## y.dl7        3.353e-01  1.273e-01   2.634 0.008560 **
## y.dl8        3.007e-01  1.226e-01   2.453 0.014332 *
## y.dl9        3.231e-01  1.178e-01   2.743 0.006192 **
## y.dl10       2.638e-01  1.129e-01   2.337 0.019619 *
## y.dl11       2.794e-01  1.078e-01   2.592 0.009675 **
## y.dl12       2.091e-01  1.031e-01   2.027 0.042872 *
## y.dl13       1.788e-01  9.779e-02   1.828 0.067748 .
## y.dl14       1.705e-01  9.272e-02   1.838 0.066254 .
## y.dl15       1.602e-01  8.707e-02   1.840 0.066015 .
## y.dl16       1.233e-01  8.121e-02   1.518 0.129285
```

```
## y.dl17       9.023e-02  7.497e-02   1.204 0.229031
## y.dl18       1.333e-01  6.803e-02   1.959 0.050319 .
## y.dl19       8.796e-02  6.032e-02   1.458 0.145083
## y.dl20       5.539e-02  5.229e-02   1.059 0.289719
## y.dl21       1.324e-02  4.266e-02   0.310 0.756332
## y.dl22       5.139e-02  2.916e-02   1.762 0.078270 .
## du           1.513e-02  4.341e-03   3.486 0.000509 ***
## dt          -1.070e-04  3.738e-05  -2.862 0.004292 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02542 on 1135 degrees of freedom
##   (23 observations effacées parce que manquantes)
## Multiple R-squared:  0.04638,    Adjusted R-squared:  0.02454
## F-statistic: 2.123 on 26 and 1135 DF,  p-value: 0.0008856
##
##
## Teststatistic: -9.1123
## Critical values: 0.01= -5.57 0.05= -5.08 0.1= -4.82
##
## Potential break point at position: 1005
```

```
# gamma(22) significatif car sa statistique t associée (1.762) est supérieure en valeur absolue au seui

# On observe que delta 1 et delta 2 sont significatif.
# En effet, leurs p-values, respectivement 0.0005 et 0.0042, sont bien
# inférieur à notre seuil de 5%, on garde donc le modele "both"
# mais beta(1) non significatif avec sa p-value de 0.7442 < 5%.

# On a donc le bon modèle, on peut donc faire le test de racine unitaire :
# H0 : rho-1 = 0
# t.stat = -9.1123  < -5.08 = t.critique ainsi on rejette H0
# Le PGD est donc TS avec date de rupture.
```

```
break_point = 1005
dates_rtt[break_point] # 2021-02-05
```

```
## [1] "2025-02-11"
```

```
# Cette date correspond à une phase de forte reprise des valeurs industrielles
# et aérospatiales en Corée du Sud, après le creux de 2020 lié à la pandémie.
# Pour Korea Aerospace Industries, cette période marque également le retour des
# commandes militaires et exportations (notamment le programme FA-50), ce qui a
# pu modifier la dynamique de la série de rendements.

# Cependant, comme le rendement n'a pas de tendance alors il ne peut pas être TS,
# le PGD est donc stationnaire avec une date de rupture le 5 février 2021.
```

Test de Lee et Strazicich (LS)

```r
# Cependant, nous pouvons rejeter H0 dans ZA alors que possibilité
# d'un PGD DS avec une date de rupture
# On va donc faire un test LS
# H0 : DS avec une date de rupture
# Ha : TS avec une date de rupture

# Comme dans ZA notre model est both alors dans LS notre model est break

# sourcer LeeStrazicichUnitRootTest.R
myLS_test <- ur.ls(y = rtt, model = "break", breaks = 1,
                   lags = 3, method = "GTOS", pn = 0.1, print.results = "print")
```

```
## [1] -37.09325
## [1] "First possible structural break at position: 404"
## [1] "The location of the first break - lambda_1: 0.3 , with the number of total observations: 1185"
## Critical values - Break model:
##      lambda    1%    5%   10%
## [1,]    0.1 -5.11 -4.50 -4.21
## [2,]    0.2 -5.07 -4.47 -4.20
## [3,]    0.3 -5.15 -4.45 -4.18
## [4,]    0.4 -5.05 -4.50 -4.18
## [5,]    0.5 -5.11 -4.51 -4.17
## [1] "Number of lags determined by general-to-specific lag selection: NA"
## Runtime:
## Time difference of 0.03257965 mins
```

```r
# t.stat = -37.09324 < -4.45  = t.critique donc on rejette H0
# Ainsi, le PGD qui a généré les données est TS avec une date de rupture.

# Nous avons suffisamment d'observation pour ne pas faire bootstrap (2213)
```

```r
break_point = 404
dates_rtt[break_point] # 2022-08-24
```

```
## [1] "2022-08-24"
```

```r
# Cette date coïncide avec la période de négociation et de finalisation du contrat
# d’exportation de 48 FA-50 vers la Pologne, annoncé fin juillet 2022 et signé en
# septembre 2022.

# Cependant, comme le rendement n'a pas de tendance alors il ne peut pas être TS,
# le PGD est donc stationnaire.
```