Mabrouk Chetouane
Université Paris Dauphine
BFT - Investment Manager

Ecole Centrale de Paris - Supelec
Séries temporelles appliquées
Cours de Pascal Bondon
03 mars 2020

Modèle $ARMA(p,q)$ et Méthode des moindres carrés ordinaires

Votre fichier devra s'intituler "NOM1 NOM2 - CENTRALE - 0320.(pdf)" et il en va de même pour l'objet du mail . La date limite d'envoi est le dimanche 8 mars à 00h00. Veuillez soigner la présentation de vos résultats ainsi que la rédaction (en Tex uniquement). Il vous est également demandé de joindre vos code dans un fichier R. L'adresse pour l'envoi du document est mabrouk.chetouane@gmail.com

Exercice : modèle d'arbitrage

Cadre : Le Fed model est un modèle empirique que l'on peut classer dans la catégorie de modèle d'arbitrage simple. Le Fed model repose sur l'idée qu'il existe une relation d'arbitrage entre la rémunération d'une obligation d'état, jugée sans risque, et le taux de rendement d'une action plus risquée. Le prolongement de cette logique induit l'existance d'une relation d'équilibre entre les deux variables. L'équation ci-dessous permettrait de décrire cette relation :

$$\frac{E_t}{P_t} = \alpha + \beta.r_t \tag{1}$$

Le modèle économétrique est donné par la relation ci-dessous :

$$\frac{E_t}{P_t} = \alpha + \beta.r_t + \epsilon_t \tag{2}$$

où E_t désigne les earnings d'un indice ou d'une entreprise, P_t le prix de cette action ou de l'indice, r_t un taux sans risque et ϵ_t le terme d'erreur du modèle et on suppose $\epsilon_t \sim \mathcal{N}\left(0, \sigma_\epsilon^2\right)$. On supposera des indices actions pour des raisons de simplicité (ici le S&P500) et nous considérons un taux de rendement d'une obligation d'état à 10 ans en guise de taux sans risque.

Partie 1 : Estimation du modèle

Question 1 - Calculer le earning yield de l'indice du S&P500. Tracer les deux courbes en faisant apparaître les dates sur l'axe des abscisses. Tracer un nuage de points liant le earning yield au taux de rendement des obligations à 10 ans. L'ajustement linéaire est-il justifier ? Utiliser la commande `"abline"` pour tracer cet ajustement.

Question 2 - Rappeler le principe de la méthode des moindres carrés ordinaires. Rappeler les hypothèses sous-jacente de l'estimateur des MCO.

Question 3 - Montrer que l'application des moindres carrés ordinaires à l'équation 2 permet de parvenir au résultat suivant:

$$\hat{\alpha} = \left(\frac{\bar{E}}{P}\right) - \hat{\beta}\bar{r} \tag{3}$$

$$\hat{\beta} = \frac{Cov\left(\frac{E}{P}, r\right)}{\sigma_r^2} \tag{4}$$

Démontrer que $lim_{T\to\infty} V\left[\hat{\beta}\right] = 0$.

Question 4 - A l'aide la fonction `"lm"` du logiciel R, estimer par les MCO les coefficients de l'équation 2. Commenter vos résultats en particulier le signe du coefficient et sa significativité. Qu'indiquent les statistiques de Student et de Fisher ainsi que le coefficient de détermination ?

Question 5 - Réaliser un étude complète des résidus estimés : tracer leur densité, étudier leur normalité et vérifier l'existence/l'absence d'autocorrélation et d'hétéscédasticité. Quel serait l'estimateur le plus adéquat si l'on venait à détecter de l'hétéroscédascité dans les résidus. Détailler votre réponse.

Question 6 - L'observation des séries et du nuage de point laisse penser qu'une rupture s'es produite. A l'aide de la commande `"window "` découper votre dataset en deux parties : depuis 1961 jusqu'à décembre 2001 puis de janvier 2002 à mars 2019. Reprsenter côte à côte les nuages de points de ses deux sous-échantillons. Que constatez-vous? Estimer le modèle décrit par l'équation 2 pour chacune des sous-périodes. Commenter vos résultats. Existe-t-il des familles de modèles capables de prendre en charge ce type de configuraion de données.

Partie 2: Estimation d'une nouvelle spécification et comparaison

Le *Fed model* comprend cependant une limite dans sa spécification. En effet, ses auteurs tentent d'expliquer le comportement d'une variable réelle $E_{i,t}/P_{i,t}$ en utilisant un taux de rendement nominal. Pour corriger cette limite, on décide de calculer un taux d'intérêt réel en déflatant ce dernier de la croissance de l'indice des prix à la consommation (CPI) sur sur 12 mois.

$$\frac{E_i}{P_i} = \alpha + \beta.(r_i - \pi_t) + \epsilon_t \tag{5}$$

$$\frac{E_t}{P_t} = \alpha + \beta.rr_t + \epsilon_t \tag{6}$$

Question 7 - Calculer le taux d'intérêt réel. Estimer par la méthode des moindres carrés ordinaires cette nouvelle spécification sur l'ensemble de la période d'une part puis sur les deux sous périodes d'autre part. Ce changement de spécification améliore-t-il le pouvoir explicatif du modèle notamment sur la seconde période? Justifier votre réponse.

Partie 3: Estimation d'une nouvelle spécification : modèle $ARMA(p,q)$

On propose de comparer ce modèle augmenté à une approche naïve basée sur un modèle $ARMA(p,q)$ sur l'ensemble de période d'étude c'es à dire depuis 1961.

Question 8 - Présenter le modèle ARMA et ses prinicpales propriétés de manière précise et succinte.

Question 9 - Identifier l'ordre du modèle ARMA à l'aide de deux méthodes différentes (on privilégiera un approche parcimonieuse). L'introduction d'une partie intégrée dans vore modèle ARMA (soit un modèle de type ARIMA) vous parait nécessaire. Pourquoi?

Question 10 - Estimer le modèle identifié à l'aide de la fonction `"auto.arima"` et vérifier la qualité de votre estimation.

Question 11 - Effectuer une prévision à l'aide de la commande `"predict"` sur horizon de trois périodes. Donner l'interval de confiance de votre prévision à 95% et tracer le sur le même graphique que votre prévision.

Une manière de sélectionner un modèle estimé consiste à évaluer sa capacité prédictive. On a alors recours à la Root Mean Square Error et de la Mean Absolute Error.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Question 12 - Comparer les deux derniers modèles estimés en utilisant ces deux critères. Quel est celui qui affiche la meilleure performance?

Question 11 bis : - Une autre manière de comparer la qualité prédictive de deux modèles économétriques consiste à mettre en oeuvre le test de Diebold et Mariano (1995). On note $r^a_{i,t+h|t}$ la prévision du taux de rendement issue du Fed model (première version) et $r^b_{i,t+h|t}$ celle issue de la seconde spécification, pour $t = t_0, \ldots, T$. Dès lors on a

$$\epsilon^a_{i,t+h|t} = r_{i,t+h} - r^a_{i,t+h|t}$$

$$\epsilon^b_{i,t+h|t} = r_{i,t+h} - r^b_{i,t+h|t}$$

On définie une fonction de perte, notée $L\left(r_{i,t+h}, r^j_{i,t+h|t}\right) = L\left(\epsilon^j_{i,t+h|t}\right)$, $avec\ j = a, b$. On retient habituellement la fonction de perte suivante $L\left(\epsilon^j_{i,t+h|t}\right) = \left(\epsilon^j_{i,t+h|t}\right)^2$. Pour savoir si un modèle est plus performant qu'un autre modèle, on teste alors l'hypothèse nulle suivante:

$$H_0 : \mathbb{E}\left[L\left(\epsilon^a_{i,t+h|t}\right)\right] = \mathbb{E}\left[L\left(\epsilon^b_{i,t+h|t}\right)\right]$$

contre l'hypothèse alternative

$$H_0 : \mathbb{E}\left[L\left(\epsilon^a_{i,t+h|t}\right)\right] \neq \mathbb{E}\left[L\left(\epsilon^b_{i,t+h|t}\right)\right]$$

Le test de Diebold et Mariano (1995) est basé sur la différence des fonctions de perte. Formellement on a,

$$d_t = L\left(\epsilon^a_{i,t+h|t}\right) - L\left(\epsilon^b_{i,t+h|t}\right)$$

L'hypothèse nulle devient alors $H_0 = \mathbb{E}[d_t] = 0$. La statistique de Diebold et Mariano (1995) est donnée par

$$S_{DM} = \frac{\bar{d}}{\sqrt{\omega}} \sim \mathcal{N}(0,1) \text{ avec } \bar{d} = \frac{1}{T_0}\sum_{t=t_0}^{T} d_j$$

$$\text{avec } \hat{\omega} = \gamma_0 + 2\sum_{j=1}^{\infty}\gamma_j \; et \; \gamma_j = cov(d_t, d_{t-j})$$

Diebold et Mariano (1995) montrent que sous l'hypothèse nulle (prévisions équivalentes) alors $S_{DM} \sim \mathcal{N}(0,1)$. On rejettera l'hypothèse nulle, au seuil de 5% si $|S_{DM}| > 1.96$.

Développer une routine sous R permettant de calculer la statistique de test puis de conclure sur la qualité prédictive du modèle.

Partie 4: Stabilité du modèle

On s'intéresse désormais à la stabilité du coefficient de l'équation du *FED model*. Plusieurs outils sont disponibles pour juger de la stabilité du modèle autrement dit des coeffcients estimés :

1. sur une estimation récursive des coefficients de l'équation. Autrement dit, on estime le modèle en ajoutant à chaque nouvelle régression une observation supplémentaire.

2. sur une estimation glissante du paramètre où l'on sélectionne une période d'estimaion que l'on décale d'une période.

Question 13 - A l'aide d'une routine que vous développez sous R, estimer les coefficients conformément à la méthode glissante. Tracer les courbes des coefficients β_i estimés ainsi que leur intervalle de confiance au seuil de 95% [1]. Commenter.

La seconde étape pour étudier la stabilité globale d'un modèle consiste à analyser les résidus issus des estimations récursives (cf. étape 1). Ces résidus sont ensuite utilisés pour calculer la statistique CUSUM qui permet de conclure en matière sur la stabilité du modèle estimé.

Question 14 - Présenter le test CUSUM. Implémenter le test de CUSUM [2] et commenter vos résultats .

Bon Courage

1. L'intervalle de confiance au niveau $\alpha = 0.95$ pour X_{t+h} peut être calculé en utilisant la fonction `"predict"` et l'option `"conf"` .

2. Pour aller plus vite, charger puis installer le package `strucchange`. Dans votre scripte, `install.packages("strucchange")` ; `library("strucchange")`. Le test CUSUM y est directement implémenté.

TP Séries temporelles appliquées

Brice Rauby, Ilyas Hanine

3/3/2020

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
## The following object is masked from 'package:base':
##
##     date
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(ggplot2)
```

Modèle ARMA(p,q) et Méthodes des moindres carrés ordinaires

PARTIE 1: Estimation du modèle

Le modèle économétrique que nous allons mettre en oeuvre est le suivant (avec les notations de l'énoncé):

$$\frac{E_t}{P_t} = \alpha + \beta \cdot r_t + \epsilon_t$$

où $\epsilon_t \sim \mathcal{N}\left(0, \sigma_\epsilon^2\right)$

Importation/traitement des données

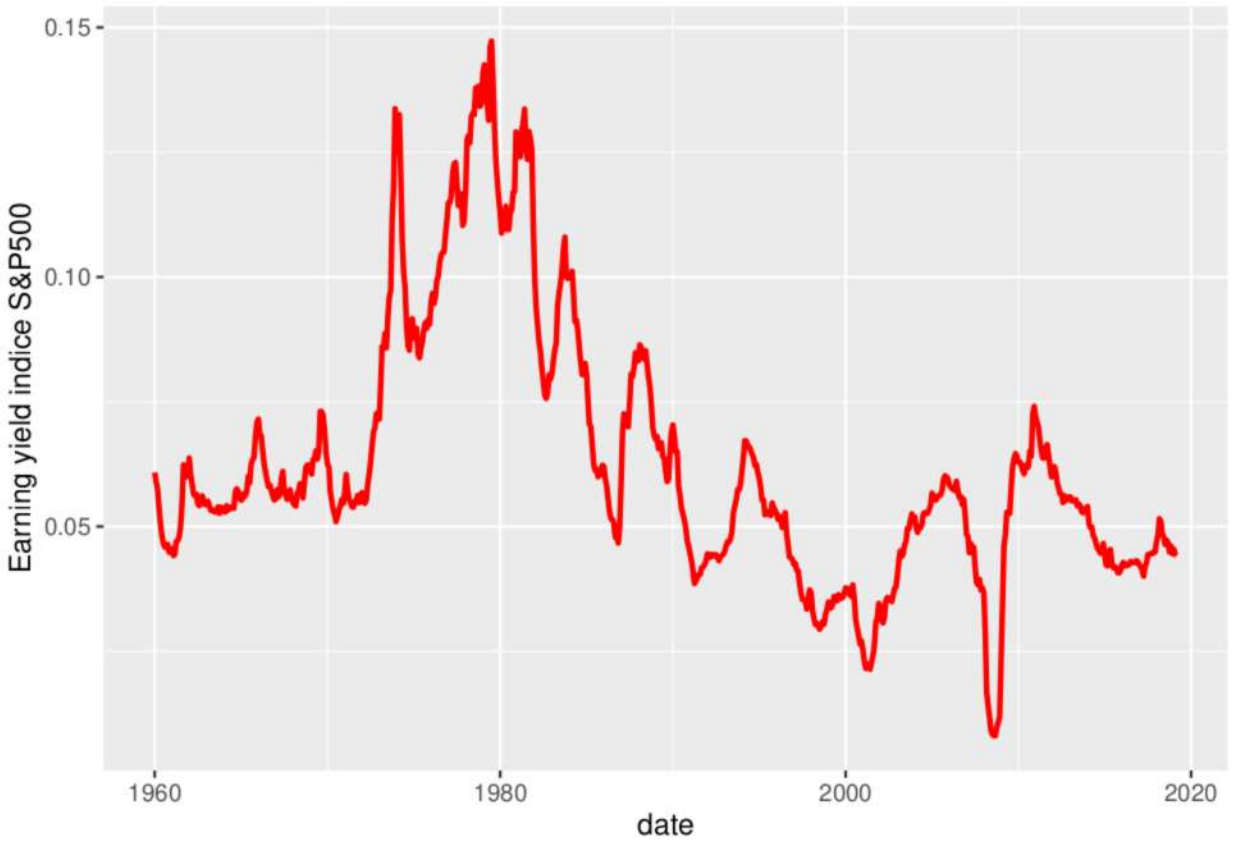
```
library(stringr)
data <- read.csv("data.csv", sep=";",encoding = "utf-8", header=TRUE)[1:711,1:6]
months=c('janv','fevr','mars','avr','mai','juin','juil','aout','sept','oct','nov','dec')
months_num=c('01','02','03','04','05','06','07','08','09','10','11','12')
data['earning yield'] <- data['earnings']/data['price']
data$X <- gsub(pattern = "\xe9" , replacement = "e" , data$X)
data$X <- gsub(pattern = "\xfb" , replacement = "u" , data$X)
data$X <- str_replace_all(data$X, months, months_num)
```

```
## Warning in stri_replace_all_regex(string, pattern,
## fix_replacement(replacement), : longer object length is not a multiple of
## shorter object length
```

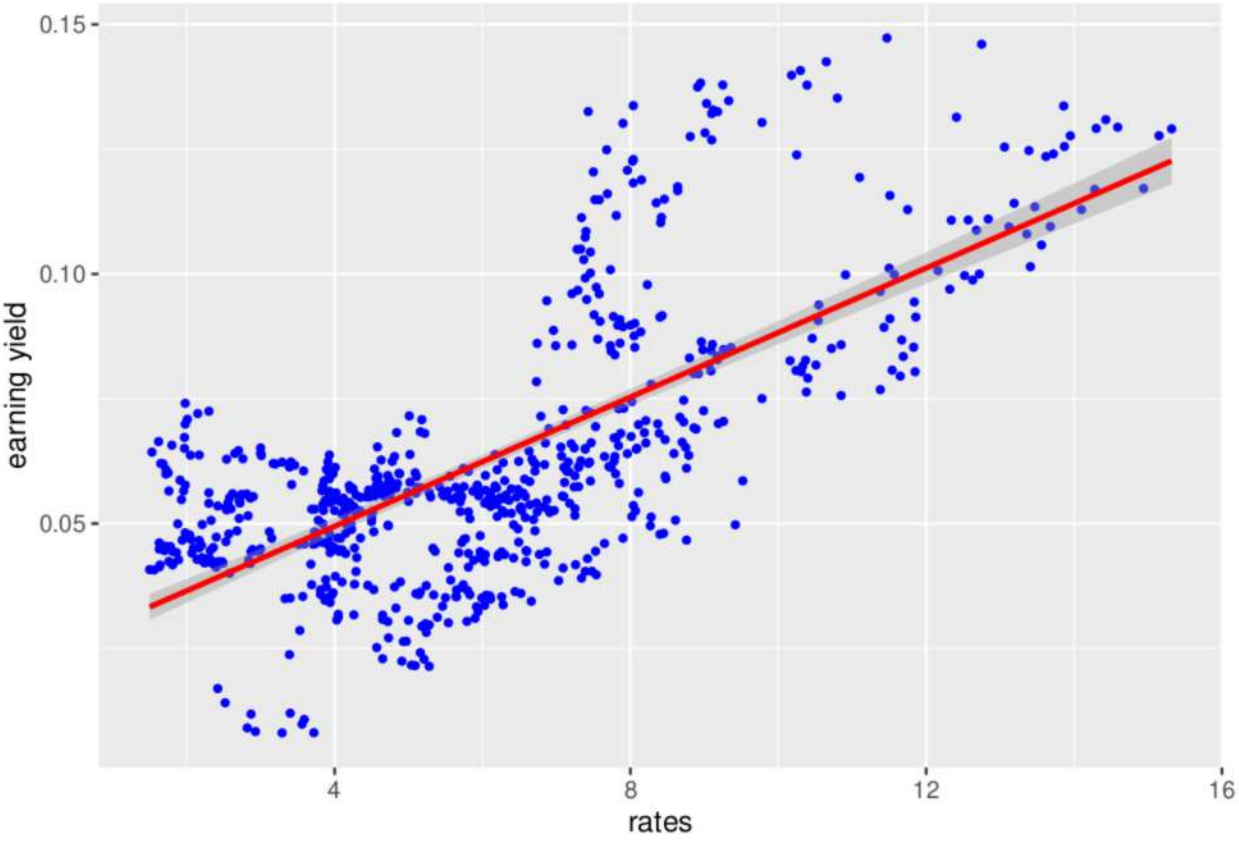
```
data$X <- as.Date(parse_date_time2(data$X,"my",cutoff_2000=20L))
setnames(data, "X", "date")
```

Tracé de la série temporelle

```
ggplot(data, aes(date, `earning yield`)) +
  geom_line(color='red',size=1)  + ylab("Earning yield indice S&P500")
```




```
ggplot(data, aes(x=rates, y=`earning yield`)) +
  geom_point(size=1, color="blue") + geom_smooth(method = "lm", se = TRUE ,color='red')
```




```
# Commande geom_smooth permet directement d'ajuster la droite de régression aux données
# Clairement la régression liénaire semble raisonnable
```

La figure permet de justifier l'intuition d'un modèle linéaire. Mettons le maintenant en oeuvre.

Question 2

Le principe de l'estimateur des moindres carrés est de mettre en place une régression linéaire au sens de la norme L_2 : Soit un ensemble d'observations $(X_i, Y_i)_{i\in[1,n]}$ (vectorielles ou scalaires).On dit que X est la variable explicative. L'estimateur des moindres carrés est donné par $\hat{\alpha}$ et $\hat{\beta}$ minimisant $\sum_{i=1}^{n} \|Y_i-(\beta X_i+\alpha)\|^2$. C'est à dire que la méthode des moindres carrés ordinaires consiste à trouver la droite qui minimise le carré de l'erreur sur l'ensemble des observations. Les hypothèses sous-jacentes sont qu'ils existent une corrélation entre X et Y et que les valeurs prises par la variable explicative sont exactes.

Question 3

Puisque l'on cherche à minimiser :

$$L(\alpha, \beta) := \sum_{i=1}^{n} \|Y_i - (\beta X_i + \alpha)\|^2$$

Dans notre cas on a $(X_i, Y_i) \in \mathbf{R}$. On note $L_i := \|Y_i - (\beta X_i + \alpha)\|^2$. La fonction étant bien convexe et coercive, cela nous permet de conclure quant à l'existence et l'unicité de du minimum dans $\mathbf{R}^2$. On cherche donc à annuler le gradient, il vient alors:

$$\frac{\partial}{\partial\alpha} L_i = \frac{\partial}{\partial\alpha}(Y_i - (\beta X_i + \alpha))^2 = -2(Y_i - (\beta X_i + \alpha))$$

$$\frac{\partial}{\partial \alpha} L = 0 \Longleftrightarrow \sum_{i=1}^{n} (Y_i - (\beta X_i + \alpha)) = 0$$

$$\frac{\partial}{\partial \alpha} L = 0 \Longleftrightarrow \sum_{i=1}^{n} (Y_i - (\beta X_i) = n\alpha$$

Ceci étant vrai pour tout β cela reste vrai pour $\hat{\beta}$, on a bien en remplaçant X_i et Y_i respectivement par r_t et $\frac{E_t}{P_t}$ et en prenant les moyennes empiriques :

$$\hat{\alpha} = \frac{1}{n} \sum_{i=1}^{n} (Y_i - (\hat{\beta} X_i)) = (\frac{\bar{E}}{P}) - \hat{\beta} \bar{r}$$

De même en dérivant par rapport à β, on obtient :

$$\frac{\partial}{\partial \beta} L_i = \frac{\partial}{\partial \beta} (Y_i - (\beta X_i + \alpha))^2 = -2X_i (Y_i - (\beta X_i + \alpha))$$

En remplaçant comme précédemment et en se plaçant en $\hat{alpha}$, on obtient :

$$\sum_{t=1}^{n} r_t (\frac{E_t}{P_t} - (\frac{\bar{E}}{P}) - \hat{\beta}(r_t - \bar{r})) = 0$$

ce qui revient à :

$$\sum_{t=1}^{n} r_t \frac{E_t}{P_t} - (\frac{\bar{E}}{P}) \sum_{t=1}^{n} r_t = \hat{\beta} (\sum_{t=1}^{n} r_t^2 - \bar{r} \sum_{t=1}^{n} r_t))$$

En remplaçant par la variance et la covariance appropriées on a bien le résultat demandé :

$$\hat{\beta} = \frac{Cov(\frac{E}{P}, r)}{\sigma_r^2}$$

Régression linéaire

```
modele <- lm(`earning yield` ~ 1 + rates, data)
summary(modele)
```

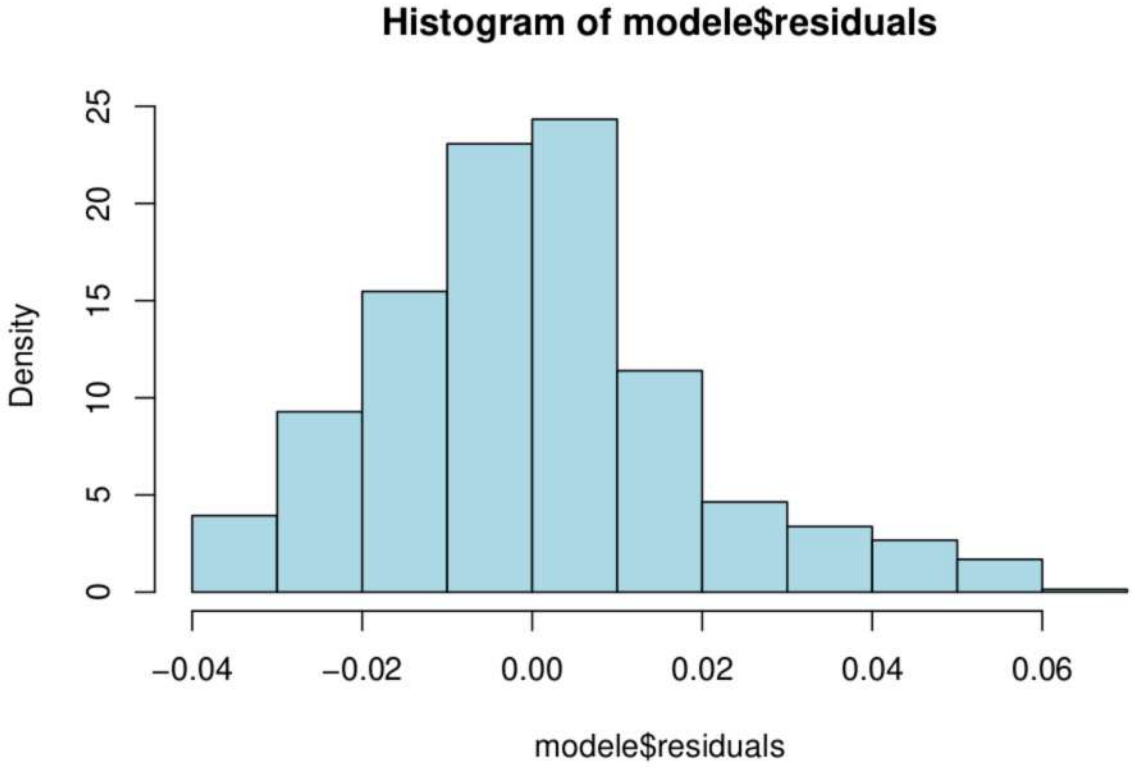
```
##
## Call:
## lm(formula = `earning yield` ~ 1 + rates, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.039533 -0.012084 -0.000824  0.009177  0.060914
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0235860  0.0016389   14.39   <2e-16 ***
## rates       0.0064668  0.0002438   26.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01875 on 709 degrees of freedom
## Multiple R-squared:  0.4981, Adjusted R-squared:  0.4974
## F-statistic: 703.6 on 1 and 709 DF,  p-value: < 2.2e-16
```

Etude des résidus

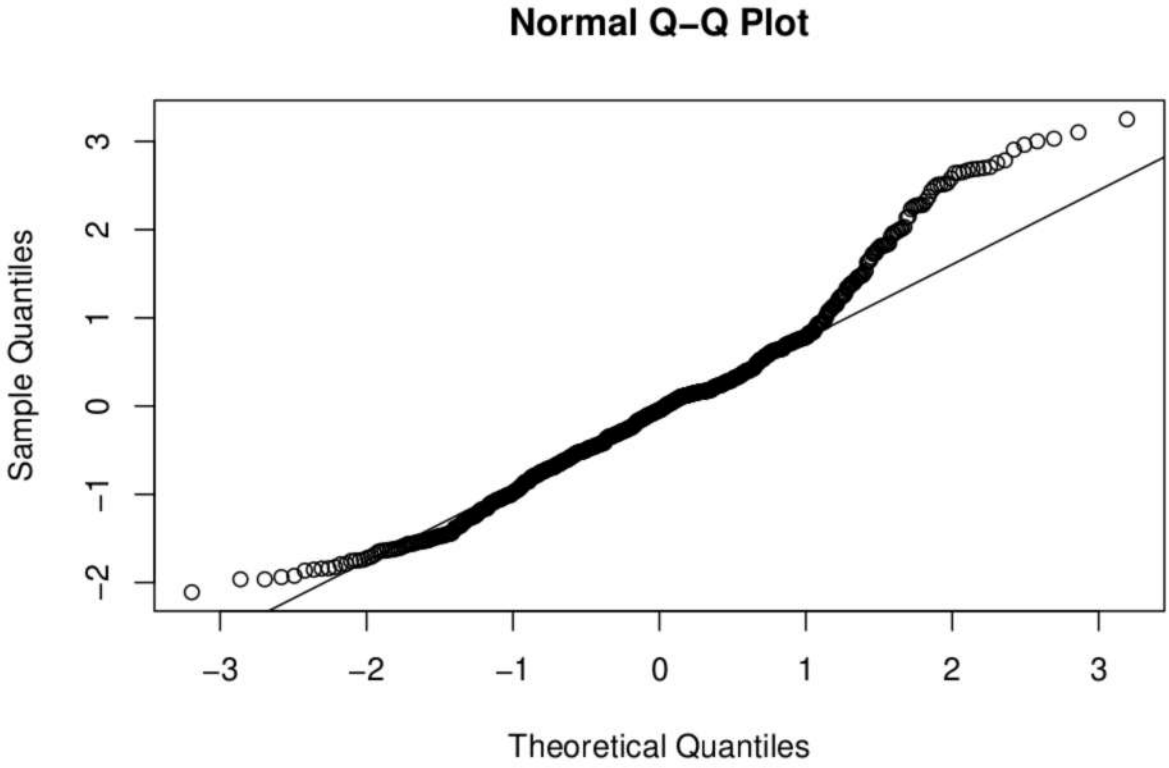
```
summary(modele$residuals)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0395333 -0.0120844 -0.0008244  0.0000000  0.0091769  0.0609139
```

```
hist(modele$residuals, freq=F, col='lightblue') #Ok pour la normalité des résidus
```



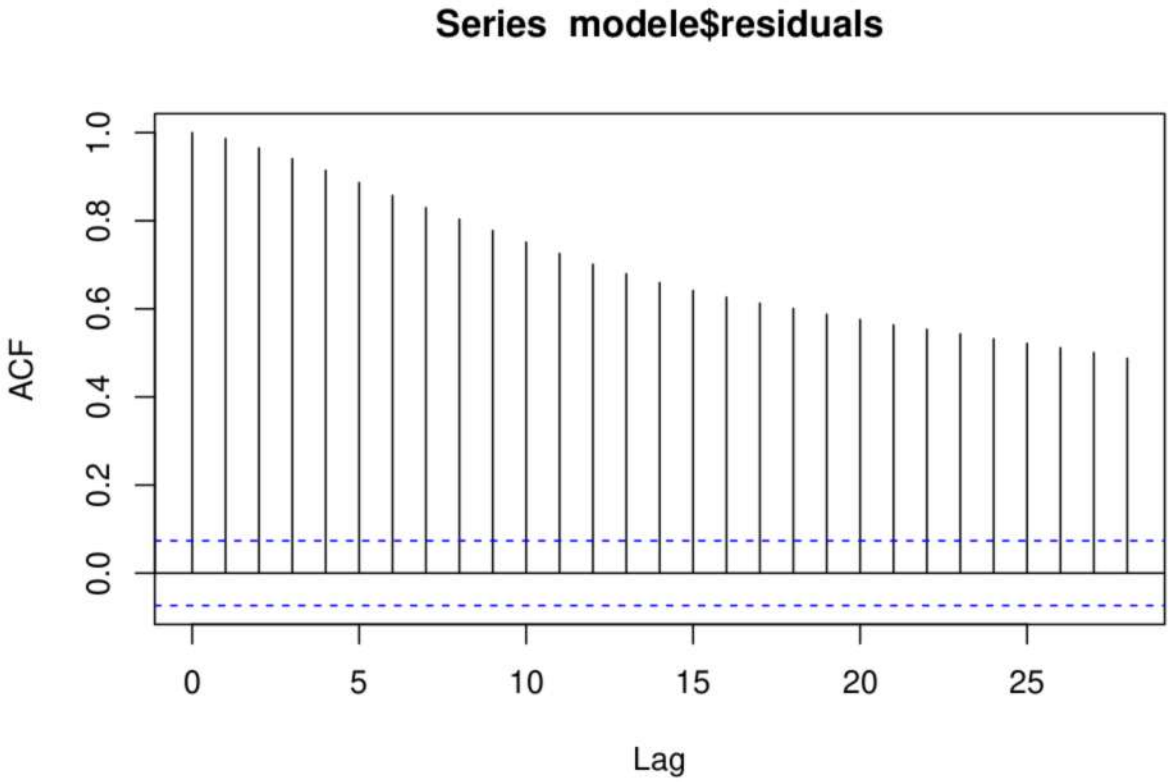

```
# Autre méthode pour vérifier la normalité des résidus :
stand_res = (modele$residuals-mean(modele$residuals))/sd(modele$residuals)
qqnorm(stand_res)
qqline(stand_res)
```



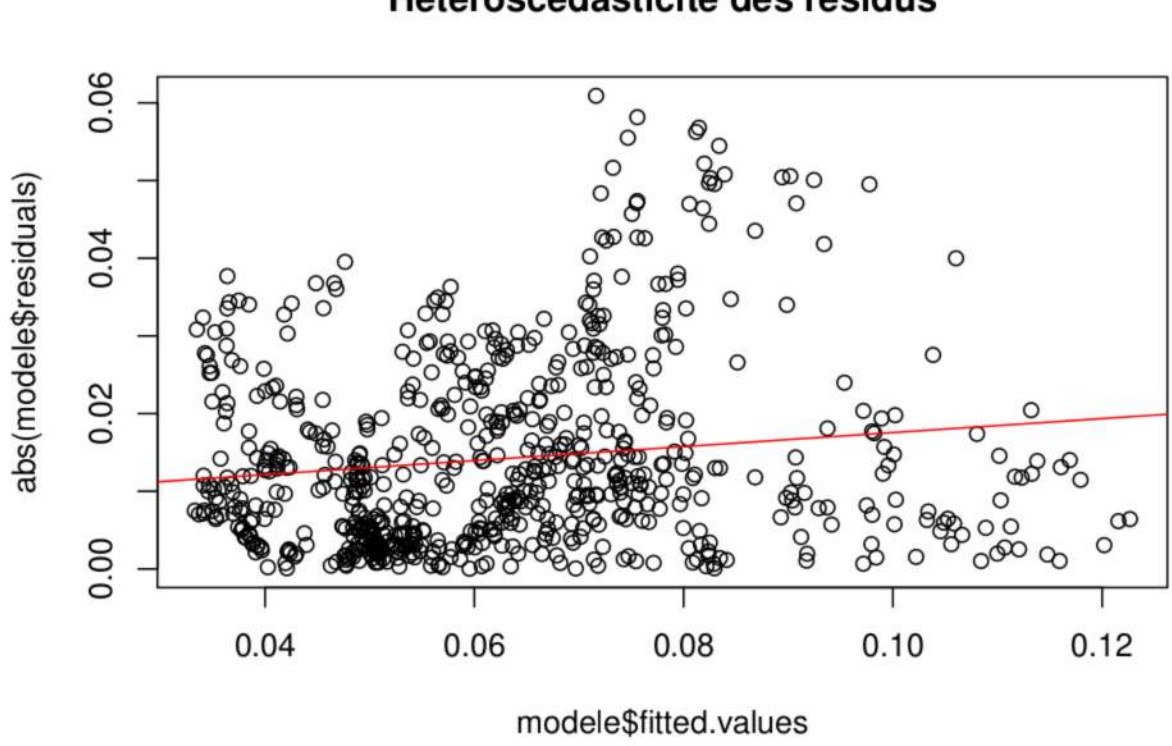

```
# Finalement on peut faire un test de normalité :
ks.test(modele$residuals,pnorm,mean=mean(modele$residuals),sd = sd(modele$residuals))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  modele$residuals
## D = 0.075955, p-value = 0.0005472
## alternative hypothesis: two-sided
```

```
# Tracé de l'autocorrélation :
acf(modele$residuals)
```




```r
# Hétéroscédasticité
plot(abs(modele$residuals) ~ modele$fitted.values)
r2 <- lm( abs(modele$residuals) ~ modele$fitted.values )
abline(r2, col="red")
title(main="Hétéroscédasticité des résidus")
```



Normalité

Le tracé de l'histogramme et de la courbe qqplot permettent raisonnablement de conclure à la normalité des données. En revanche les conclusions sont loin d'être parfaites.

Autocorrélation

Les résidus semblent très fortement corrélés au vue du tracé de l'ACF. On conclut à l'existence d'une relation linéaire entre les résidus.

Hétéroscédasticité

L'hétéroscédasticité des résidus signifie l'absence de relation entre la variance des résidus et les valeurs de la variable prédite. Ceci ne semble pas tout à fait vérifié d'après le tracé. En effet la ligne rouge de variance semble augmenter à mesure que les valeurs de variables prédites augmentent.

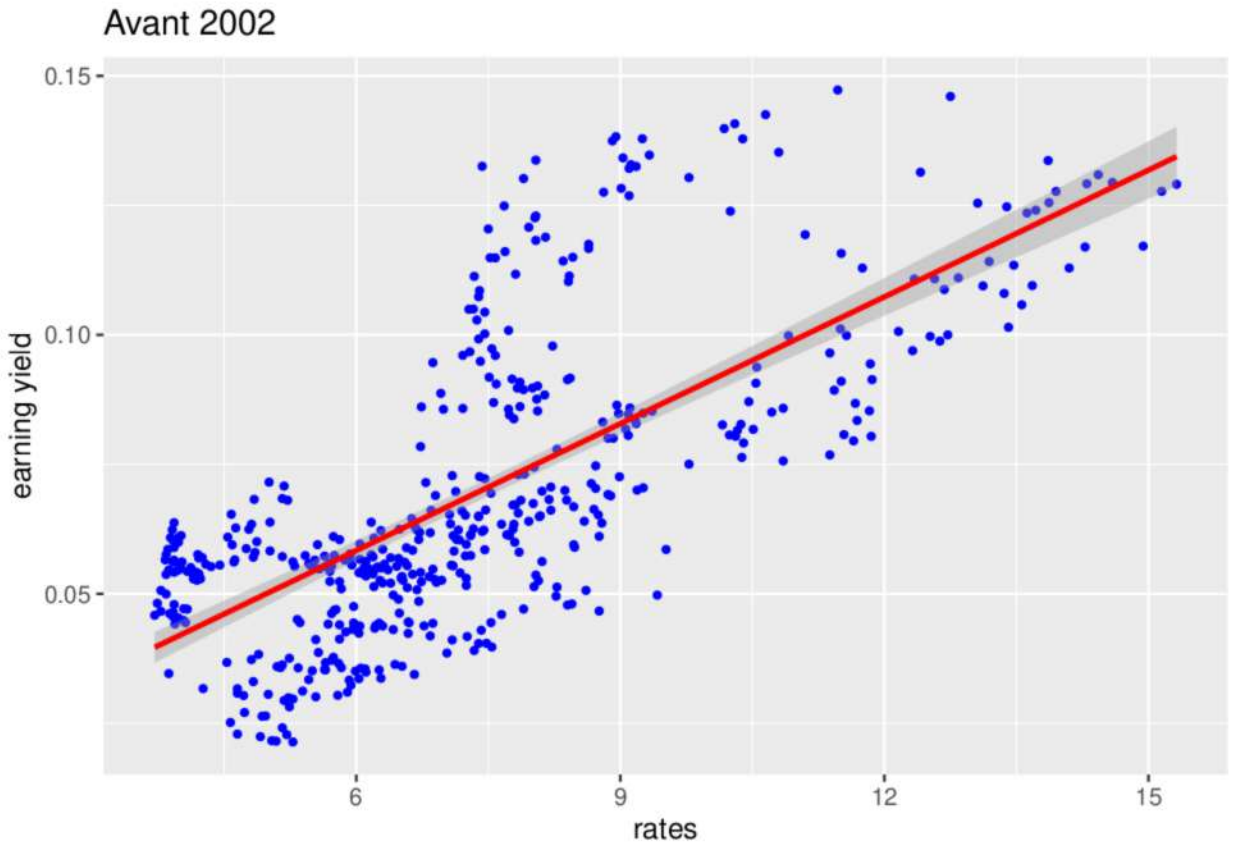
Séparation dataset

On découpe ici notre dataset en deux parties: depuis 1961 jusqu'à décembre 2001 puis de janvier 2002 à mars 2019. On déroule ensuite la même analyse que précédemment.

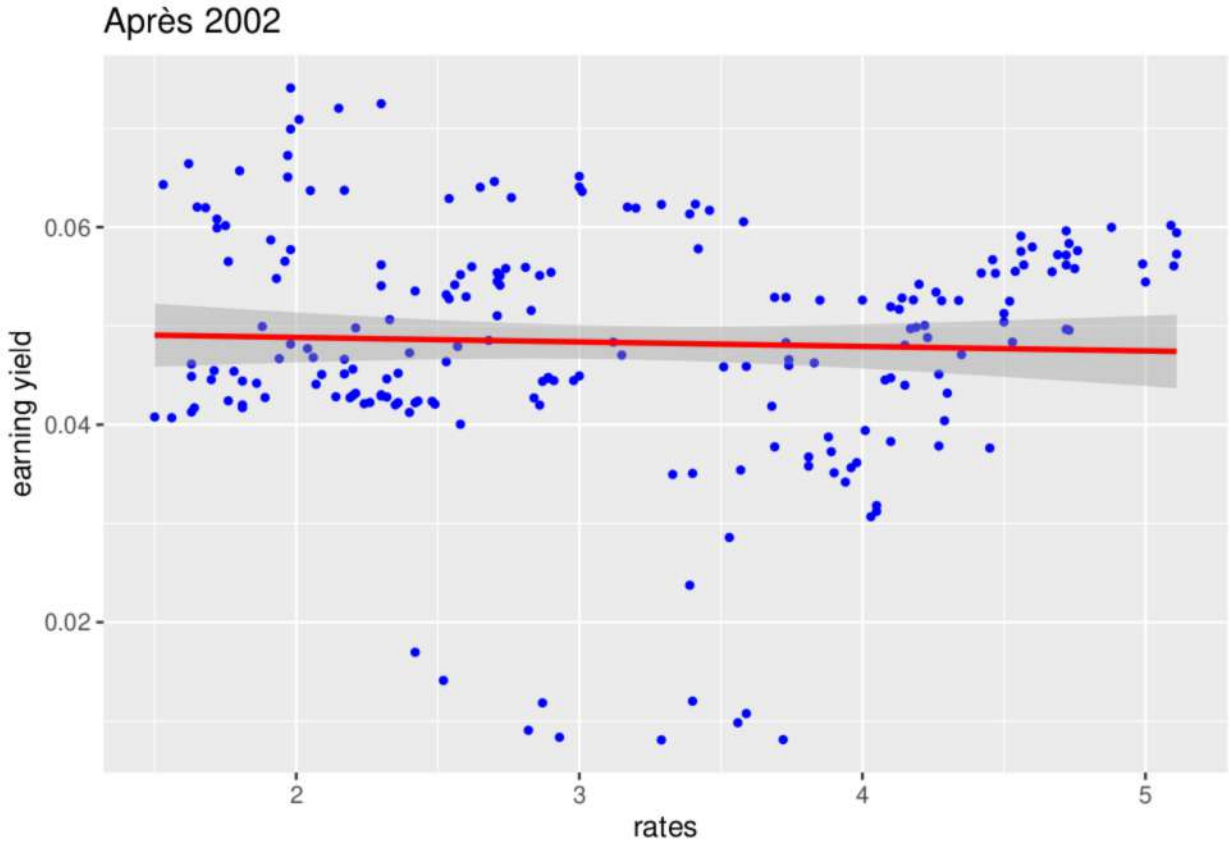
Ce qui change particulièrement, c'est lors de la régression pour la période à partir de 2002 l'absence d'information contenu dans le taux qui se traduit par la nullité du coefficient β (nullité vérifiée par le test de Student, la *t-value* étant faible).

Les résidus ne sont plus corrélés à partir d'un certain rang et cette fois nous obtenons bien l'hétéroscédasticité des résidus.

```
lst <-split(data,data$date<as.Date("2002-01-01"))
data_before=data.frame(lst[2])
data_after=data.frame(lst[1])
colnames(data_after) <- colnames(data)
colnames(data_before) <- colnames(data)
ggplot(data_before, aes(x=rates, y=`earning yield`)) + geom_point(size=1, color="blue") +
  geom_smooth(method = "lm", se = TRUE ,color='red') + ggtitle("Avant 2002")
```




```
ggplot(data_after, aes(x=rates, y=`earning yield`)) + geom_point(size=1, color="blue") +
  geom_smooth(method = "lm", se = TRUE ,color='red') + ggtitle("Après 2002")
```



MCO Avant 2002

```
modele.before <- lm(`earning yield` ~ 1 + rates, data_before)
summary(modele.before)
```

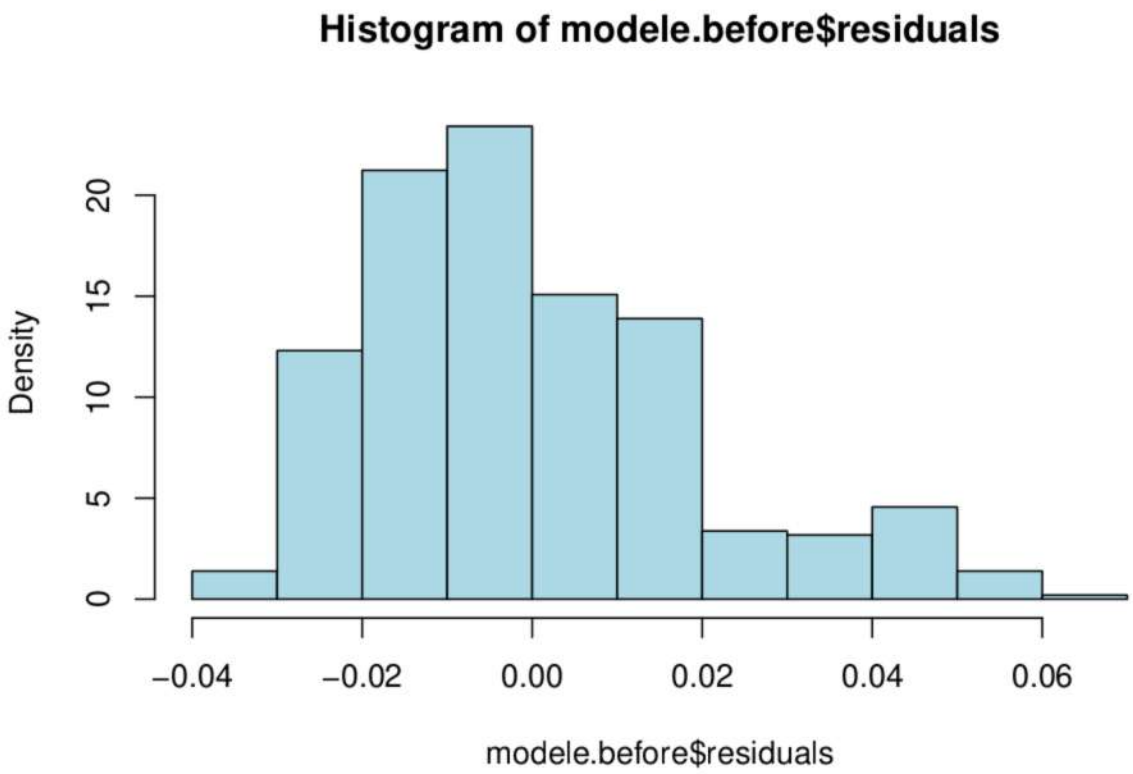
```
##
## Call:
## lm(formula = `earning yield` ~ 1 + rates, data = data_before)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.036507 -0.013412 -0.003798  0.010548  0.062518
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0093908  0.0026783   3.506 0.000495 ***
## rates       0.0081615  0.0003478  23.468  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01958 on 502 degrees of freedom
## Multiple R-squared:  0.5232, Adjusted R-squared:  0.5222
## F-statistic: 550.8 on 1 and 502 DF,  p-value: < 2.2e-16
```

Etudes des résidus

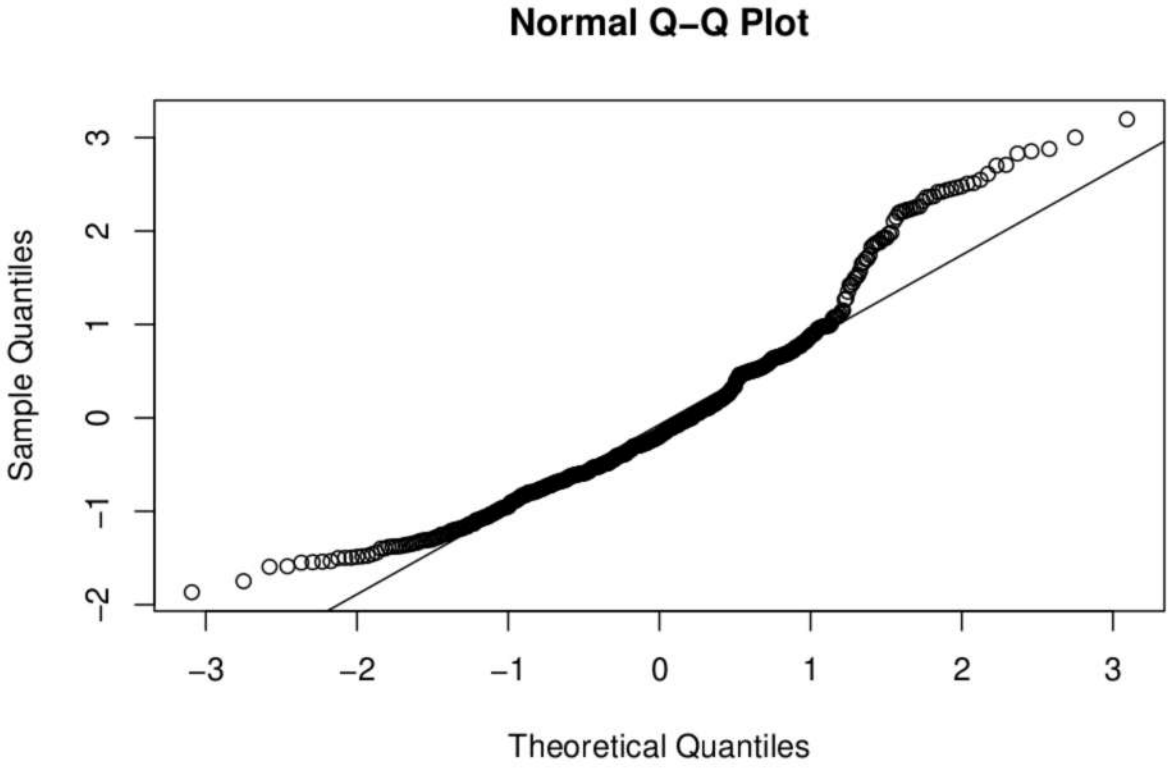
```
summary(modele.before$residuals) #résidus centrés OK
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.036507 -0.013412 -0.003798  0.000000  0.010548  0.062518
```

```
hist(modele.before$residuals, freq=F, col='lightblue')
```



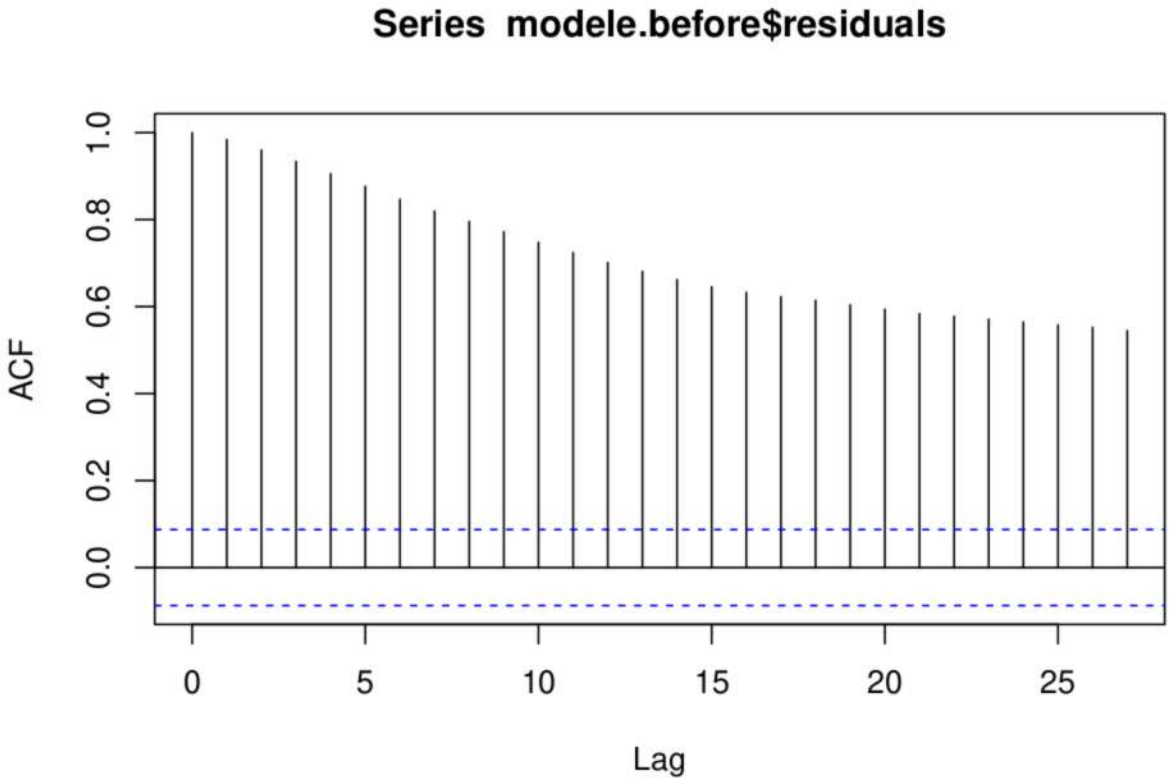

```
# Autre méthode pour vérifier la normalité des résidus :
stand_res.before =
  (modele.before$residuals-mean(modele.before$residuals))/sd(modele.before$residuals)
qqnorm(stand_res.before)
qqline(stand_res.before)
```



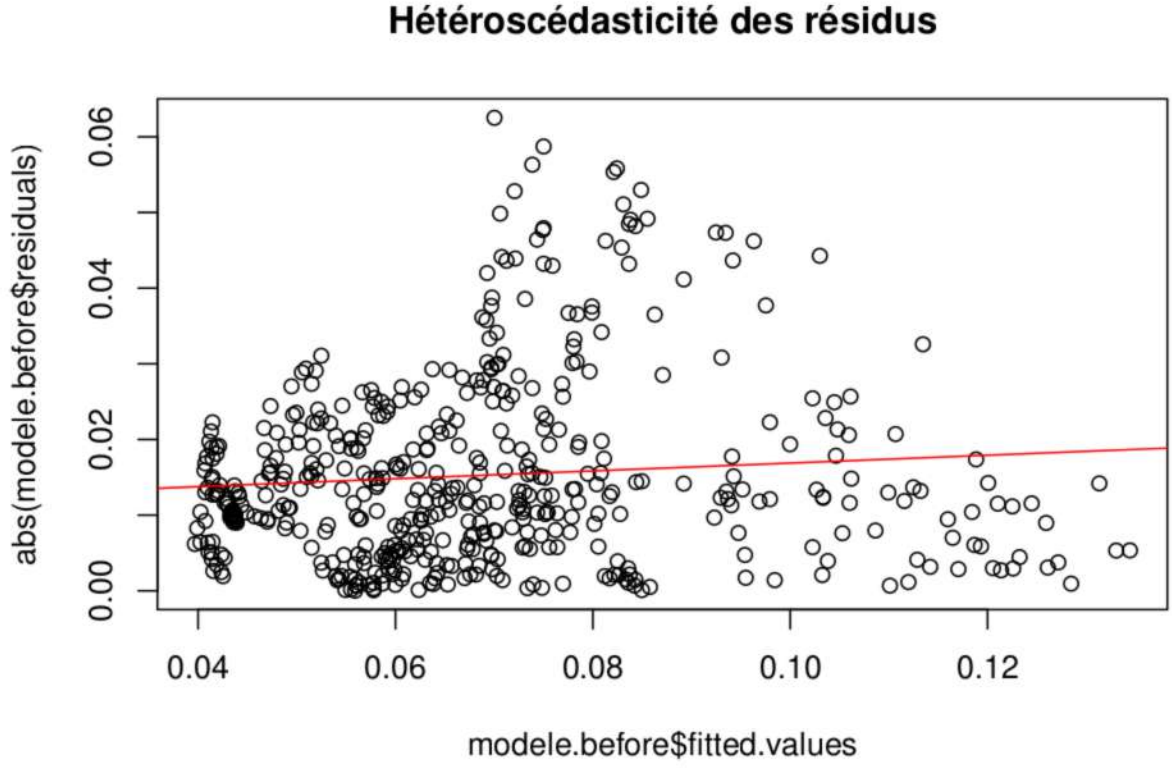

```
# Finalement on peut faire un test de normalité :
ks.test(modele.before$residuals,pnorm,mean=mean(modele.before$residuals),
        sd = sd(modele.before$residuals))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  modele.before$residuals
## D = 0.086744, p-value = 0.001016
## alternative hypothesis: two-sided
```

```
# Tracé de l'autocorrélation :
acf(modele.before$residuals)
```




```r
# Hétéroscédasticité
plot(abs(modele.before$residuals) ~ modele.before$fitted.values)
r2 <- lm( abs(modele.before$residuals) ~ modele.before$fitted.values )
abline(r2, col="red")
title(main="Hétéroscédasticité des résidus")
```



MCO après 2002

```
modele.after <- lm(`earning yield` ~ 1 + rates, data_after)
summary(modele.after)
```

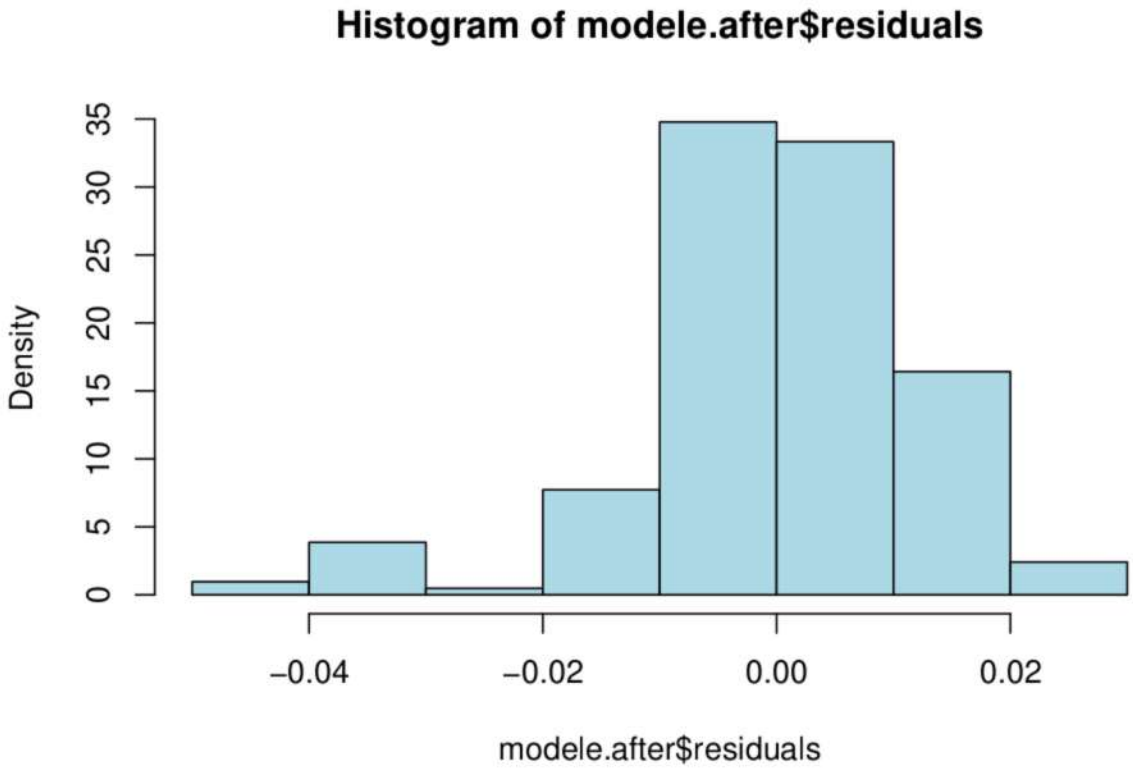
```
##
## Call:
## lm(formula = `earning yield` ~ 1 + rates, data = data_after)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.040174 -0.006078  0.000674  0.008040  0.025248
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0497502  0.0027822  17.881   <2e-16 ***
## rates       -0.0004553  0.0008459  -0.538    0.591
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0124 on 205 degrees of freedom
## Multiple R-squared:  0.001411,   Adjusted R-squared:  -0.00346
## F-statistic: 0.2897 on 1 and 205 DF,  p-value: 0.591
```

Etudes des résidus

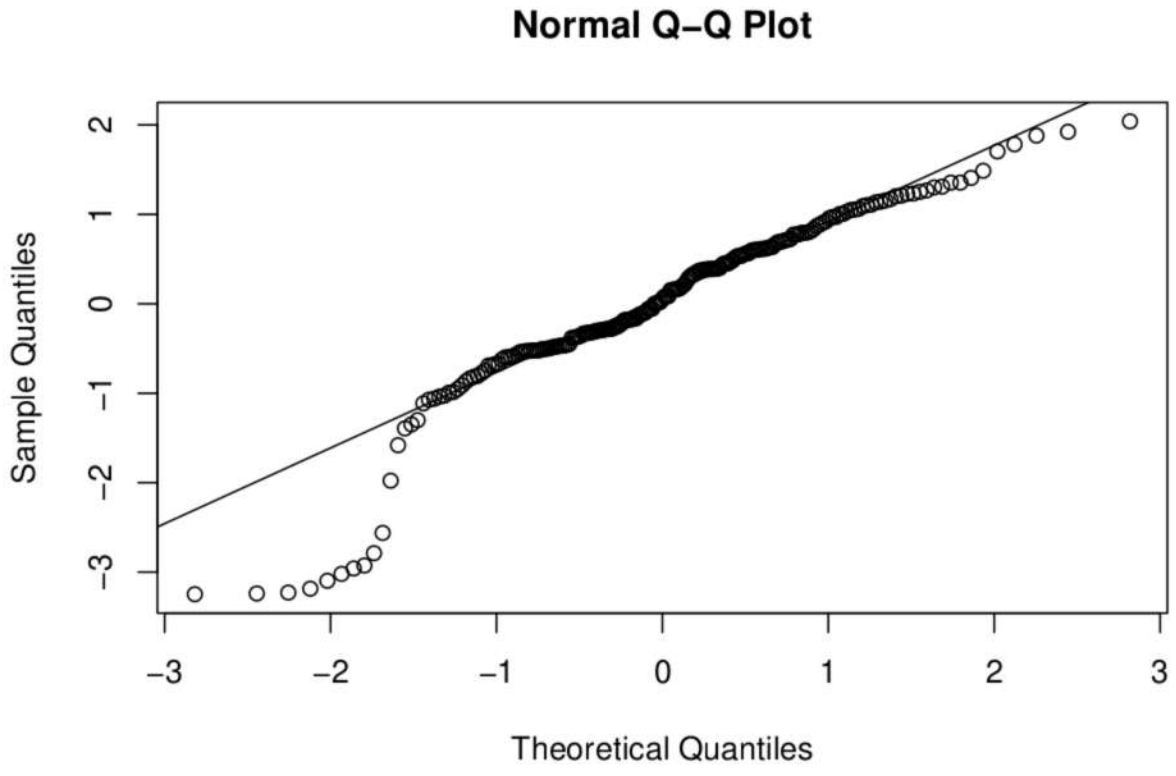
```
summary(modele.after$residuals)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0401738 -0.0060777  0.0006738  0.0000000  0.0080399  0.0252475
```

```
hist(modele.after$residuals, freq=F, col='lightblue')
```



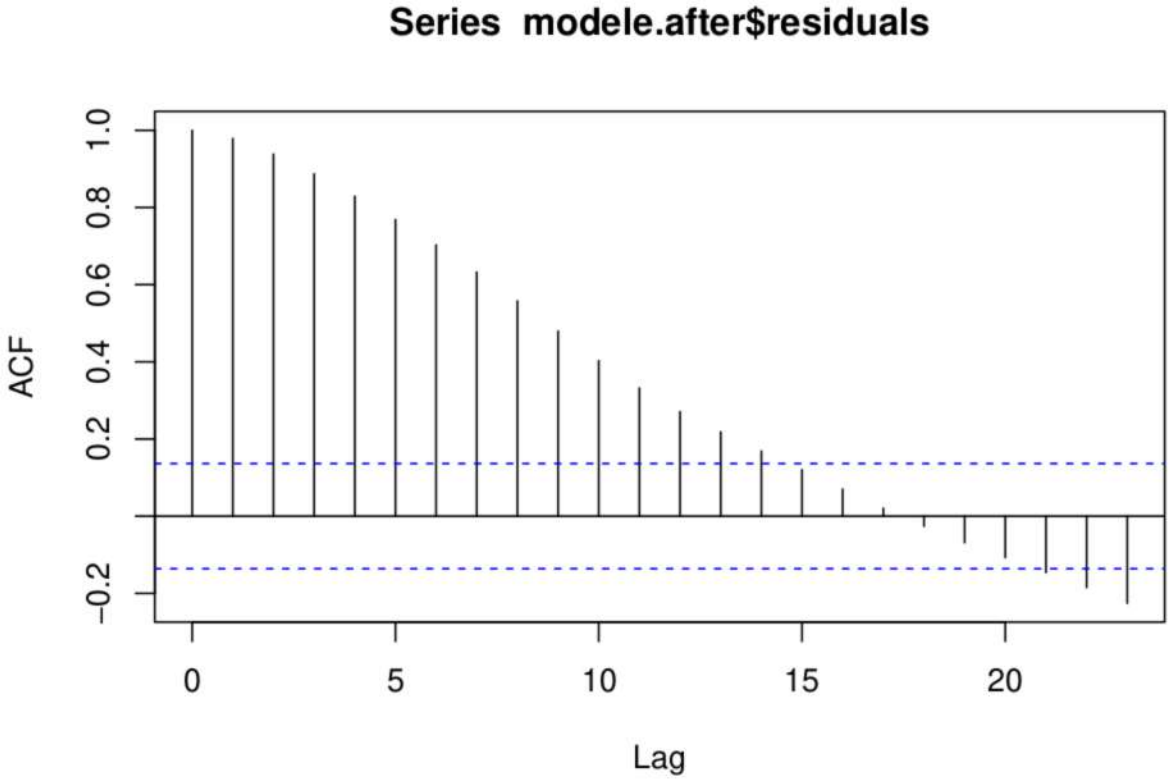

```r
# Autre méthode pour vérifier la normalité des résidus :
stand_res.after =
  (modele.after$residuals-mean(modele.after$residuals))/sd(modele.after$residuals)
qqnorm(stand_res.after)
qqline(stand_res.after)
```



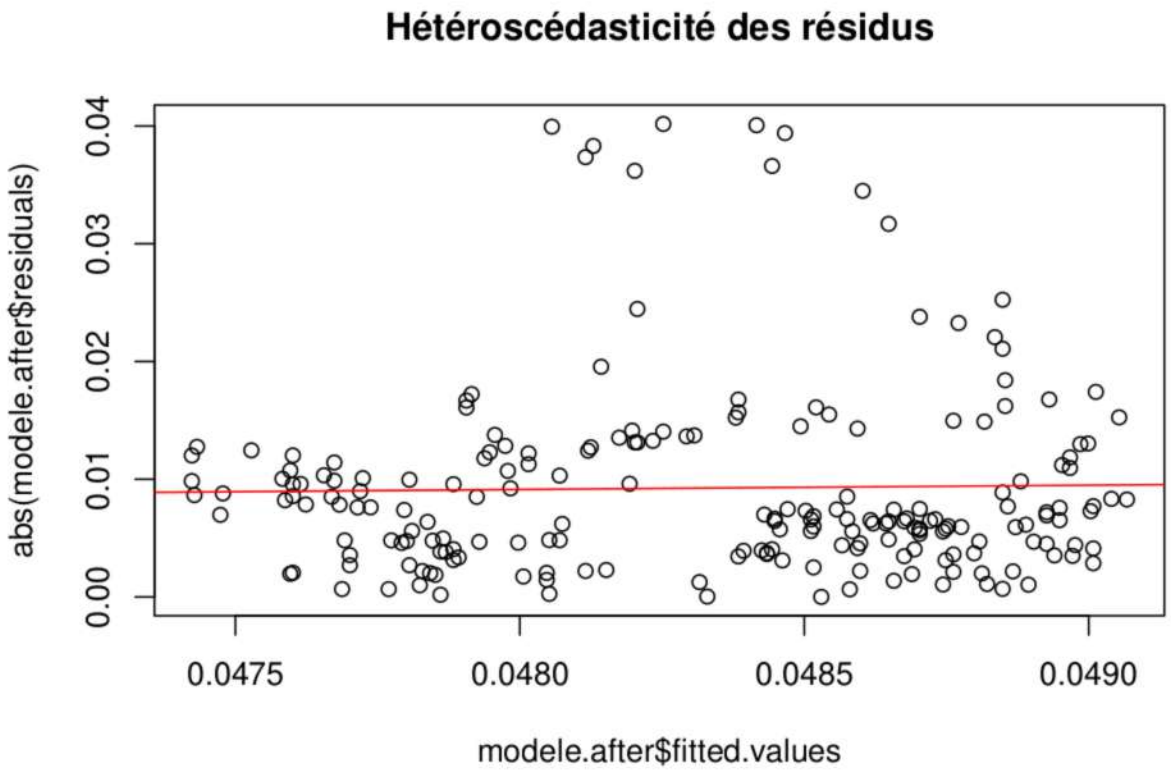
```r
# Finalement on peut faire un test de normalité :
ks.test(modele.after$residuals,pnorm,mean=mean(modele.after$residuals),
        sd = sd(modele.after$residuals))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  modele.after$residuals
## D = 0.10535, p-value = 0.02021
## alternative hypothesis: two-sided
```

```r
# Tracé de l'autocorrélation :
acf(modele.after$residuals)
```




```r
# Hétéroscédasticité
plot(abs(modele.after$residuals) ~ modele.after$fitted.values)
r2 <- lm( abs(modele.after$residuals) ~ modele.after$fitted.values )
abline(r2, col="red")
title(main="Hétéroscédasticité des résidus")
```



PARTIE 2 : Estimation d'une nouvelle spécification et comparaison

Le modèle est ici modifié de l'évolution du CPI sur l'année courante :

$$\frac{E_t}{P_t} = \alpha' + \beta' \cdot (r_t - \pi_{année}) + \epsilon_t$$

En appliquant les MCO, on espère pouvoir ainsi trouver un taux r'_t réel :

$$\frac{E_t}{P_t} = \alpha' + \beta' \cdot rr_t + \epsilon_t = \alpha' + \beta' \cdot r'_t + \epsilon_t$$

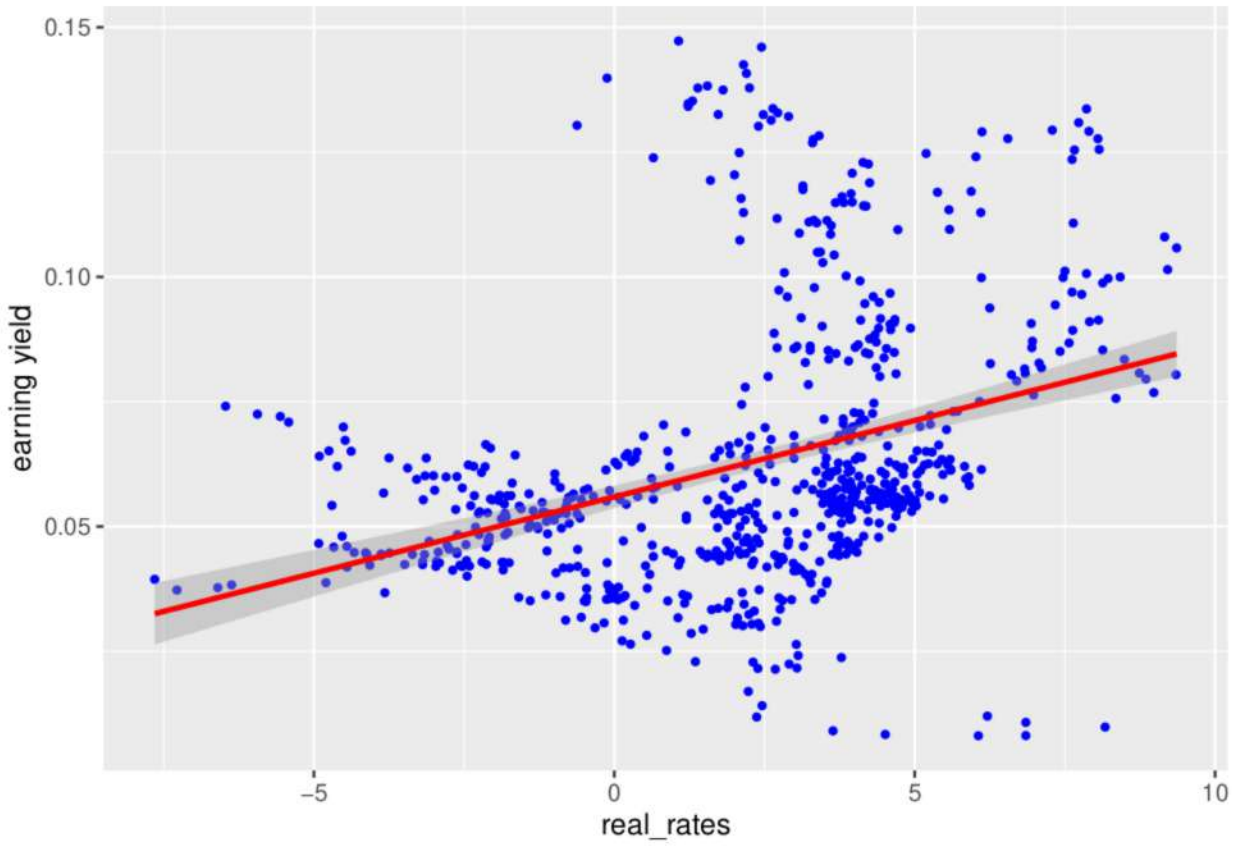
Jeu de données complet pour le modèle corrigé

```
data[1:12,'real_rates']=data[1:12,'rates']
for(i in 13:711){
  data[i,'real_rates']=data[i,'rates']-(data[i,'cpi']-data[i-12,'cpi'])
}
modele.mci <- lm(`earning yield` ~ real_rates, data)
summary(modele.mci)
```

```
##
## Call:
## lm(formula = `earning yield` ~ real_rates, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.071115 -0.014852 -0.004702  0.007747  0.088073
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0559340  0.0011523   48.54   <2e-16 ***
## real_rates  0.0030602  0.0003022   10.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02474 on 709 degrees of freedom
## Multiple R-squared:  0.1264, Adjusted R-squared:  0.1251
## F-statistic: 102.6 on 1 and 709 DF,  p-value: < 2.2e-16
```

```
ggplot(data, aes(x=real_rates, y=`earning yield`)) + geom_point(size=1, color="blue") +
  geom_smooth(method = "lm", se = TRUE ,color='red')
```



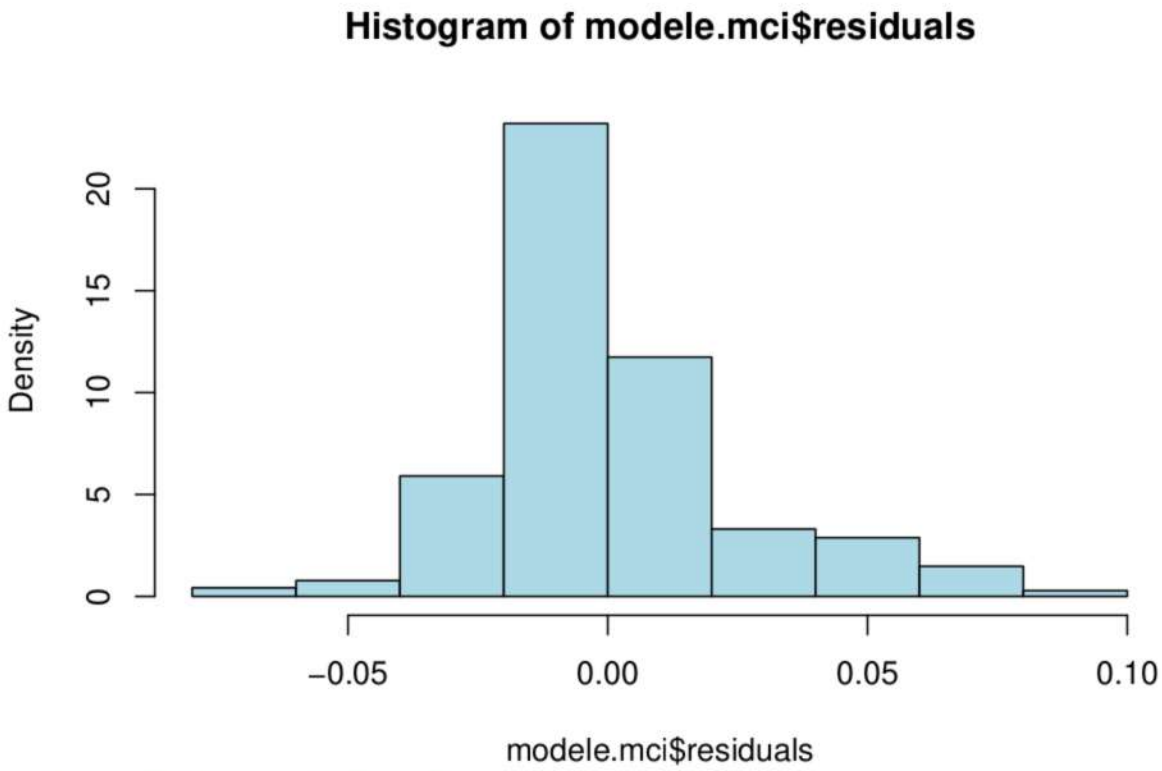
Le coefficient de détermination de 12% (inférieur à ce que l'on avait initialement) montre que la régression est moins adaptée. Nous pouvons réaliser une étude des résidus du modèle obtenu, mais avant même de les visualiser nous savons qu'ils seront mauvais.

Etude des résidus

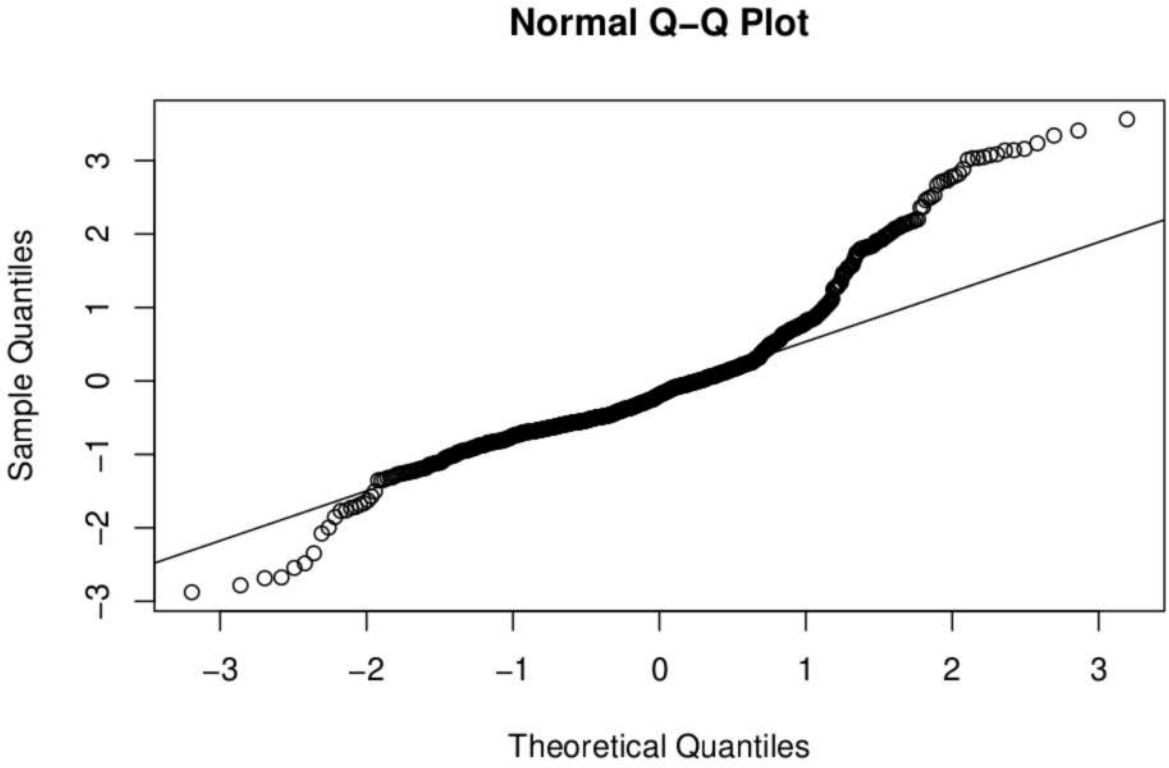
```
summary(modele.mci$residuals)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.071115 -0.014852 -0.004702  0.000000  0.007747  0.088073
```

```r
hist(modele.mci$residuals, freq=F, col='lightblue')
```



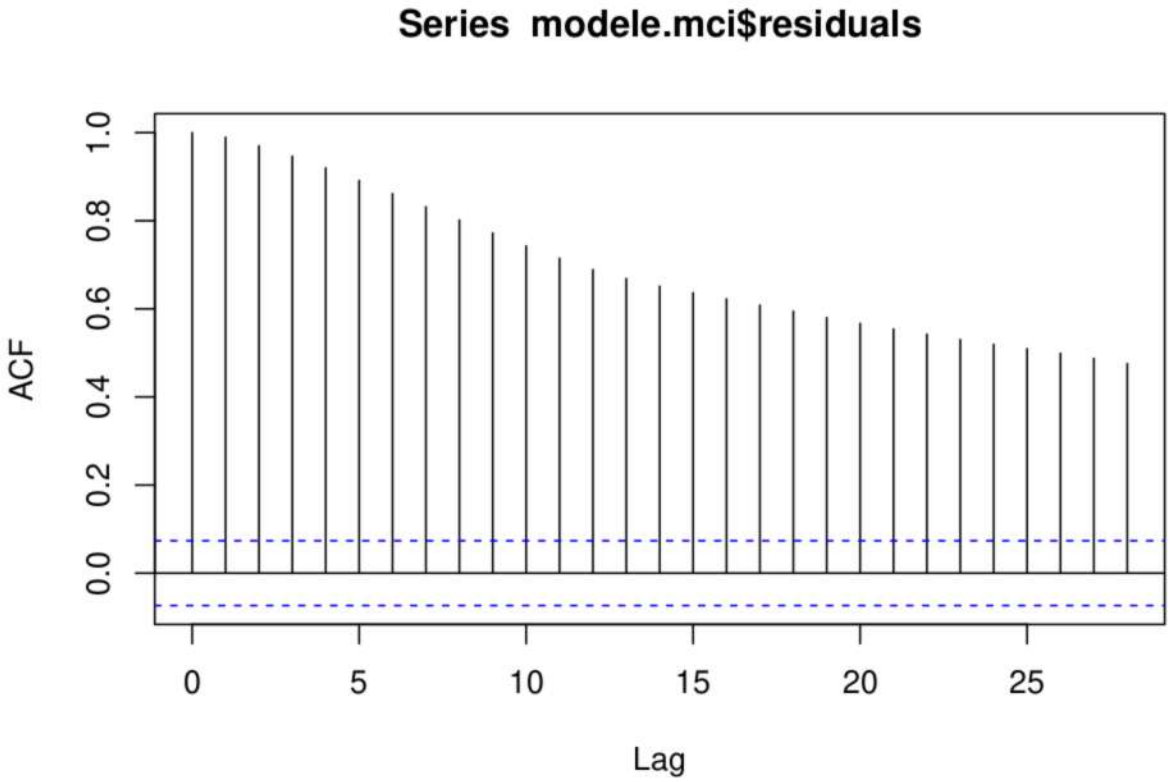

```r
# Autre méthode pour vérifier la normalité des résidus :
stand_res.mci =
  (modele.mci$residuals-mean(modele.mci$residuals))/sd(modele.mci$residuals)
qqnorm(stand_res.mci)
qqline(stand_res.mci)
```



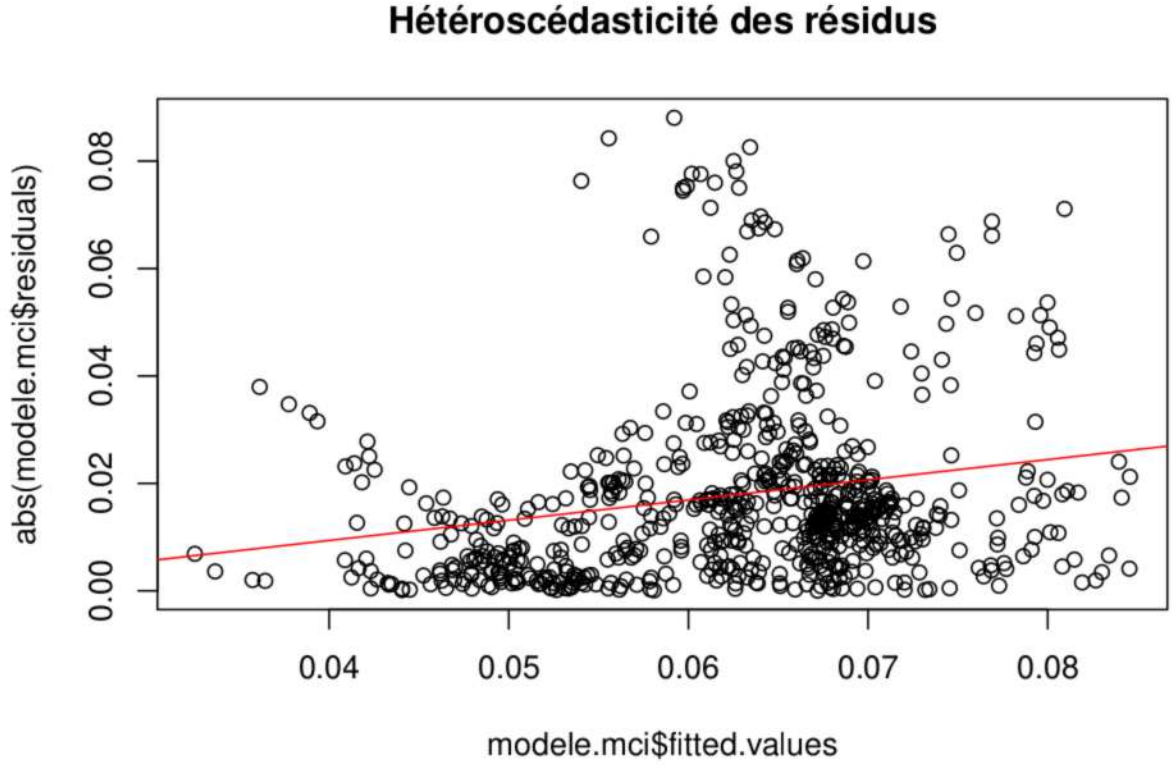

```
# Finalement on peut faire un test de normalité :
ks.test(modele.mci$residuals,pnorm,mean=mean(modele.mci$residuals),
        sd = sd(modele.mci$residuals))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  modele.mci$residuals
## D = 0.13649, p-value = 6.254e-12
## alternative hypothesis: two-sided
```

```
# Tracé de l'autocorrélation :
acf(modele.mci$residuals)
```




```
# Hétéroscédasticité
plot(abs(modele.mci$residuals) ~ modele.mci$fitted.values)
r2 <- lm( abs(modele.mci$residuals) ~ modele.mci$fitted.values )
abline(r2, col="red")
title(main="Hétéroscédasticité des résidus")
```



L'études des résidus ne vient que confirmer ce qui a été dit précédemment, à savoir que le modèle corrigé n'est pas adapté. (Pas d'hétéroscédasticité, une normalité pas vraiment vérifiée etc.)

Appliquons maintenant le modèle corrigé aux deux sous-jeux de données.

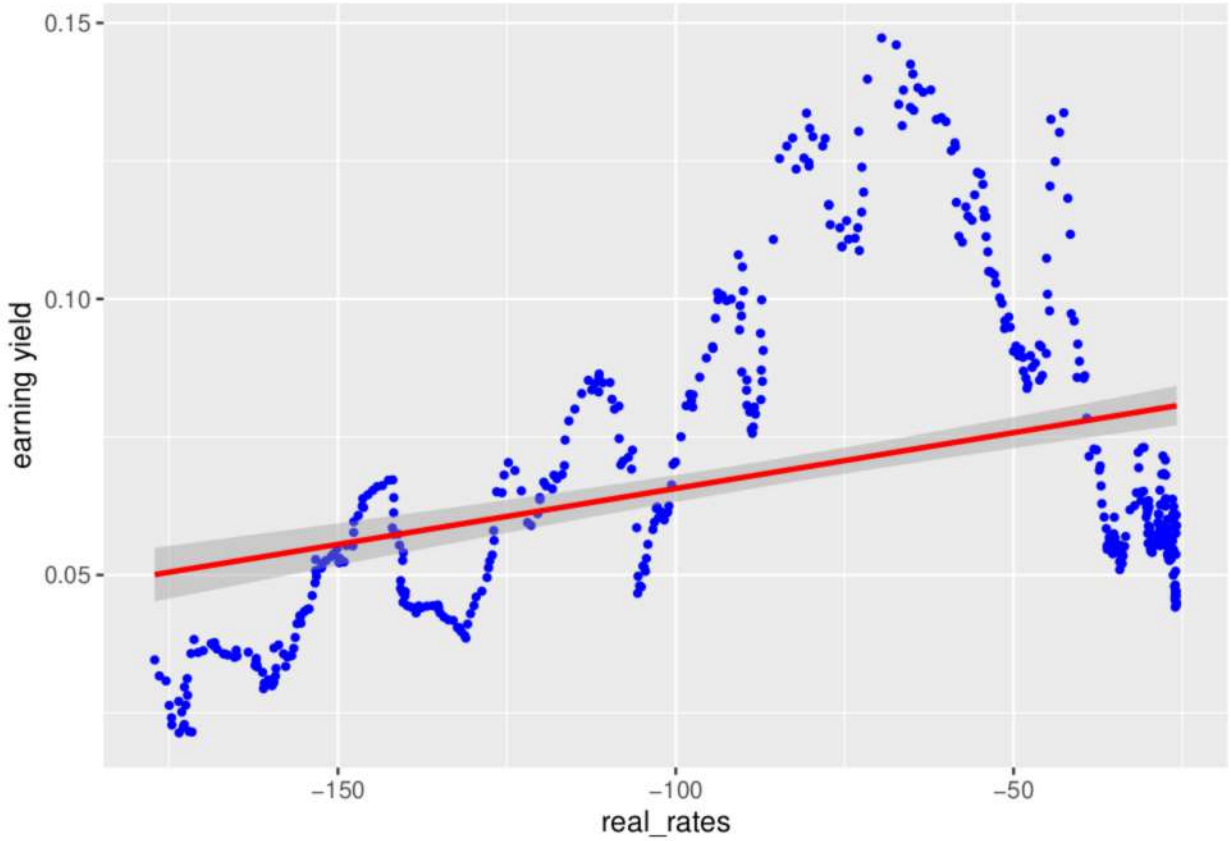
MCO avant 2002 pour le modèle corrigé

```
data_before[1:12,'real_rates']=data_before[1:12,'rates']
for(i in 13:504){
  data_before[i,'real_rates']=
    data_before[i,'rates']-(data_before[i,'cpi']-data_before[i-12,'cpi'])
}
```

```
data_before['real_rates']=data_before$rates-data_before$cpi
modele.mci.before <- lm(`earning yield` ~ real_rates, data_before)
summary(modele.mci.before)
```

```
##
## Call:
## lm(formula = `earning yield` ~ real_rates, data = data_before)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.036449 -0.021269 -0.009328  0.015179  0.075451
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.590e-02  2.305e-03  37.270   <2e-16 ***
## real_rates  2.023e-04  2.346e-05   8.623   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02646 on 502 degrees of freedom
## Multiple R-squared:  0.129,  Adjusted R-squared:  0.1273
## F-statistic: 74.36 on 1 and 502 DF,  p-value: < 2.2e-16
```

```
ggplot(data_before, aes(x=real_rates, y=`earning yield`)) + geom_point(size=1, color="blue") +
  geom_smooth(method = "lm", se = TRUE ,color='red')
```



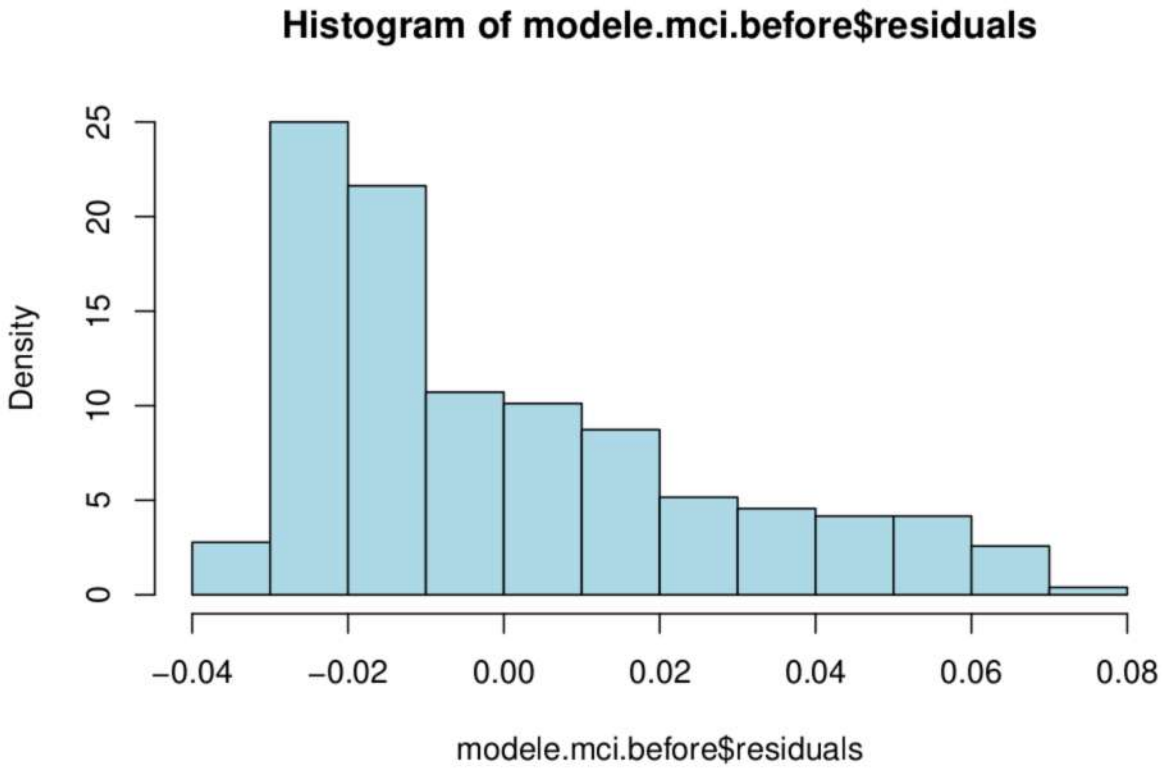
Le coefficient de détermination est toujours faible (12%), étudions maintenant les résidus du modèle. Rappelons nous que ce qui justifie le modèle corrigé (ie ce qui nous intéresse) doit se passer après 2002.

Etudes des résidus

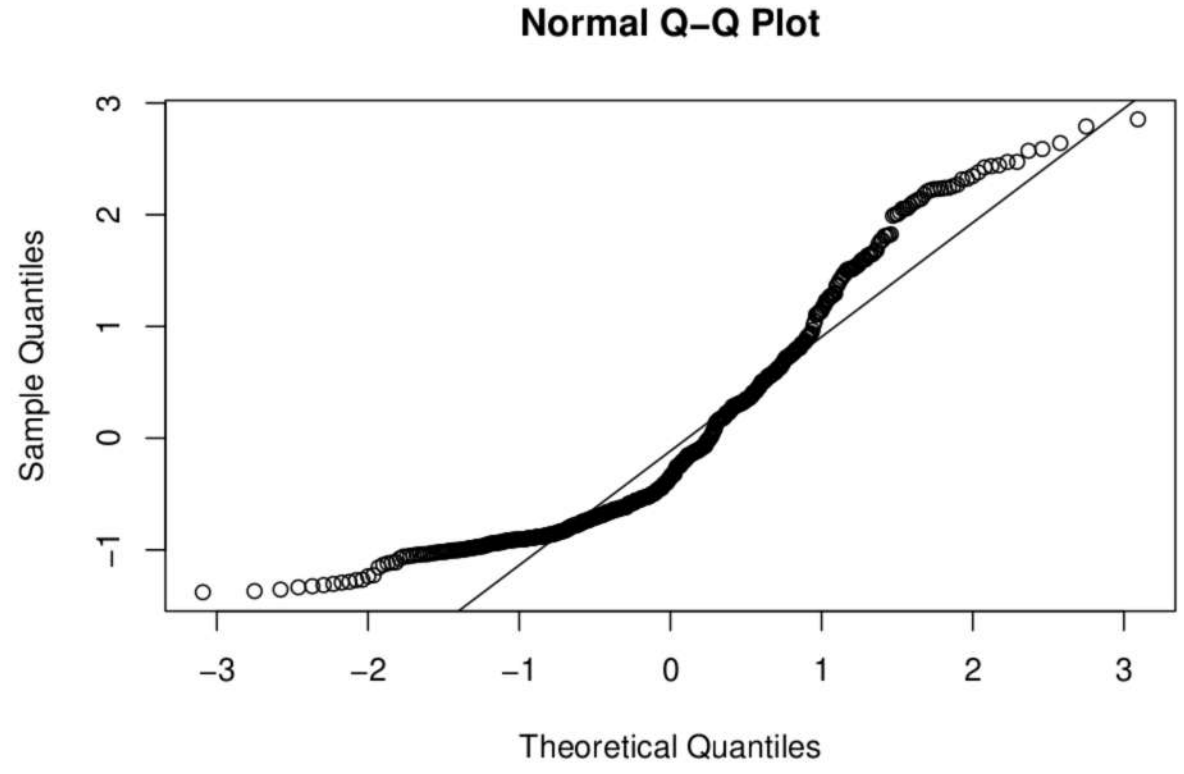
```
summary(modele.mci.before$residuals)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.036449 -0.021269 -0.009328  0.000000  0.015179  0.075451
```

```
hist(modele.mci.before$residuals, freq=F, col='lightblue')
```



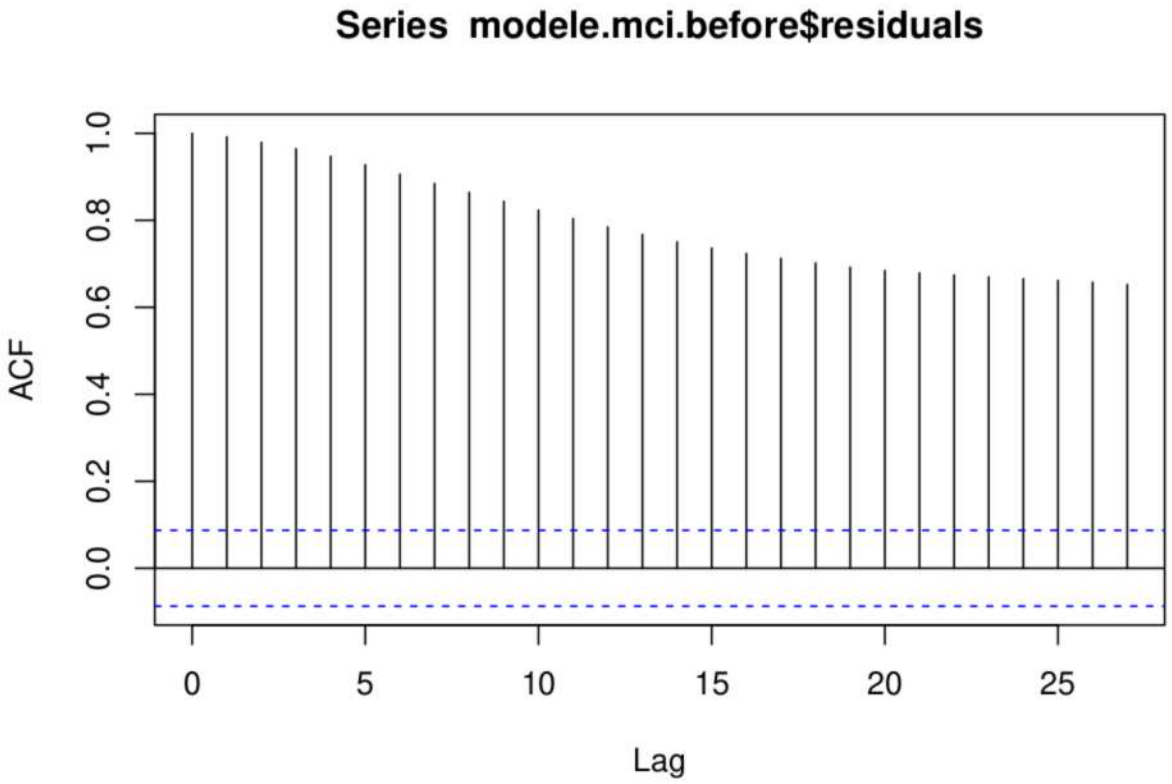

```r
# Autre méthode pour vérifier la normalité des résidus :
stand_res.mci.before =
  (modele.mci.before$residuals-mean(modele.mci.before$residuals))/sd(modele.mci.before$residuals)
qqnorm(stand_res.mci.before)
qqline(stand_res.mci.before)
```



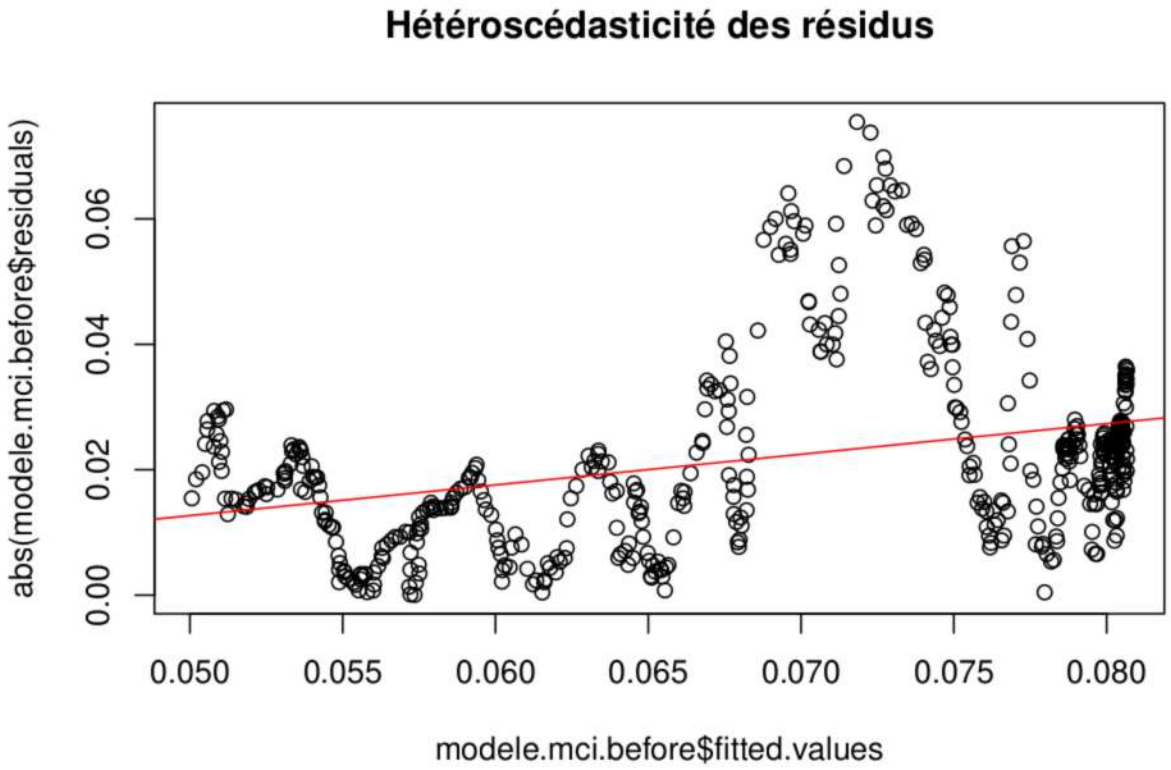
```r
# Finalement on peut faire un test de normalité :
ks.test(modele.mci.before$residuals,pnorm,mean=mean(modele.mci.before$residuals),
        sd = sd(modele.mci.before$residuals))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  modele.mci.before$residuals
## D = 0.1502, p-value = 2.662e-10
## alternative hypothesis: two-sided
```

```r
# Tracé de l'autocorrélation :
acf(modele.mci.before$residuals)
```




```r
# Hétéroscédasticité
plot(abs(modele.mci.before$residuals) ~ modele.mci.before$fitted.values)
r2 <- lm( abs(modele.mci.before$residuals) ~ modele.mci.before$fitted.values )
abline(r2, col="red")
title(main="Hétéroscédasticité des résidus")
```



La densité n'a pas plus rien de gaussienne et l'hétéroscédasticité n'est plus vérifiée. Mais encore une fois rappelons nous que le modèle corrigé est censé trouver sa justification dans les données d'après 2002. Vérifions cela dans ce qui suit immédiatement.

MCO après 2002 pour le modèle corrigé

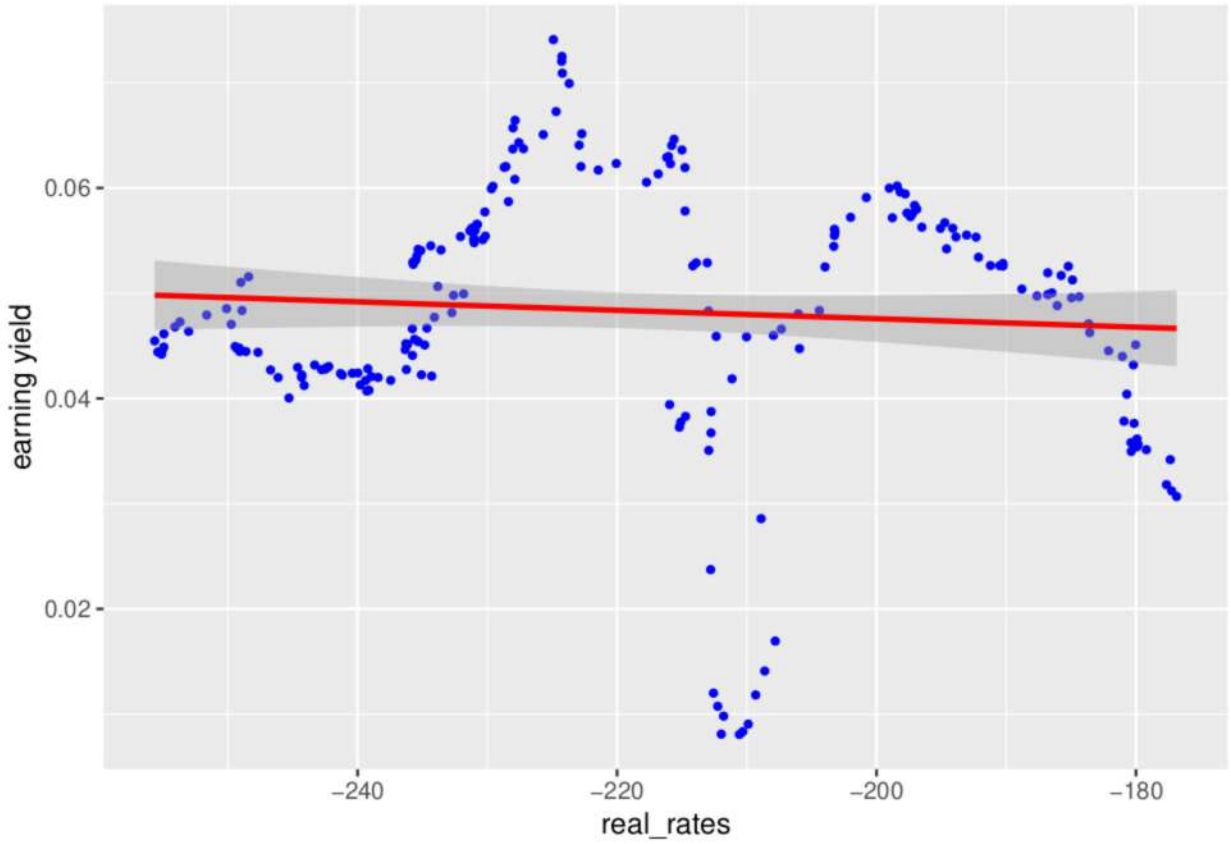
```
data_after[1:12,'real_rates']=data_after[1:12,'rates']
for(i in 13:207){
  data_after[i,'real_rates']=
    data_after[i,'rates']-(data_after[i,'cpi']-data_after[i-12,'cpi'])
}
```

```
data_after['real_rates']=data_after$rates-data_after$cpi
modele.mci.after <- lm(`earning yield` ~ real_rates, data_after)
summary(modele.mci.after)
```

```
##
## Call:
## lm(formula = `earning yield` ~ real_rates, data = data_after)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.039954 -0.006452  0.000601  0.008159  0.025513
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.956e-02  8.486e-03   4.662 5.65e-06 ***
## real_rates  -4.012e-05  3.863e-05  -1.039      0.3
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01238 on 205 degrees of freedom
## Multiple R-squared:  0.005234,   Adjusted R-squared:  0.0003814
## F-statistic: 1.079 on 1 and 205 DF,  p-value: 0.3002
```

```
ggplot(data_after, aes(x=real_rates, y=`earning yield`)) + geom_point(size=1, color="blue") +
  geom_smooth(method = "lm", se = TRUE ,color='red')
```

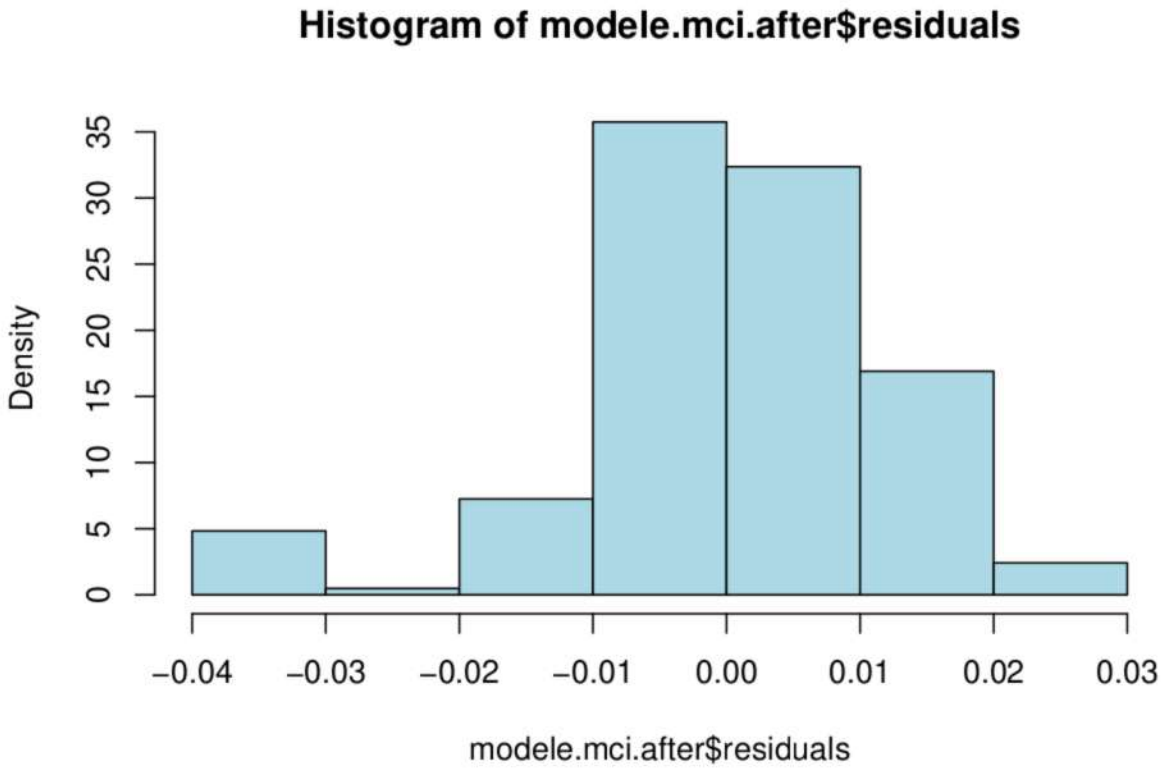


Le coefficient β affilié au taux réel est ici significatif (le test de Student permet de conclure). Comme prévu le modèle corrigé permet donc bien d'améliorer le pouvoir explicatif du modèle.
Voyons ce qu'il en est pour les résidus.

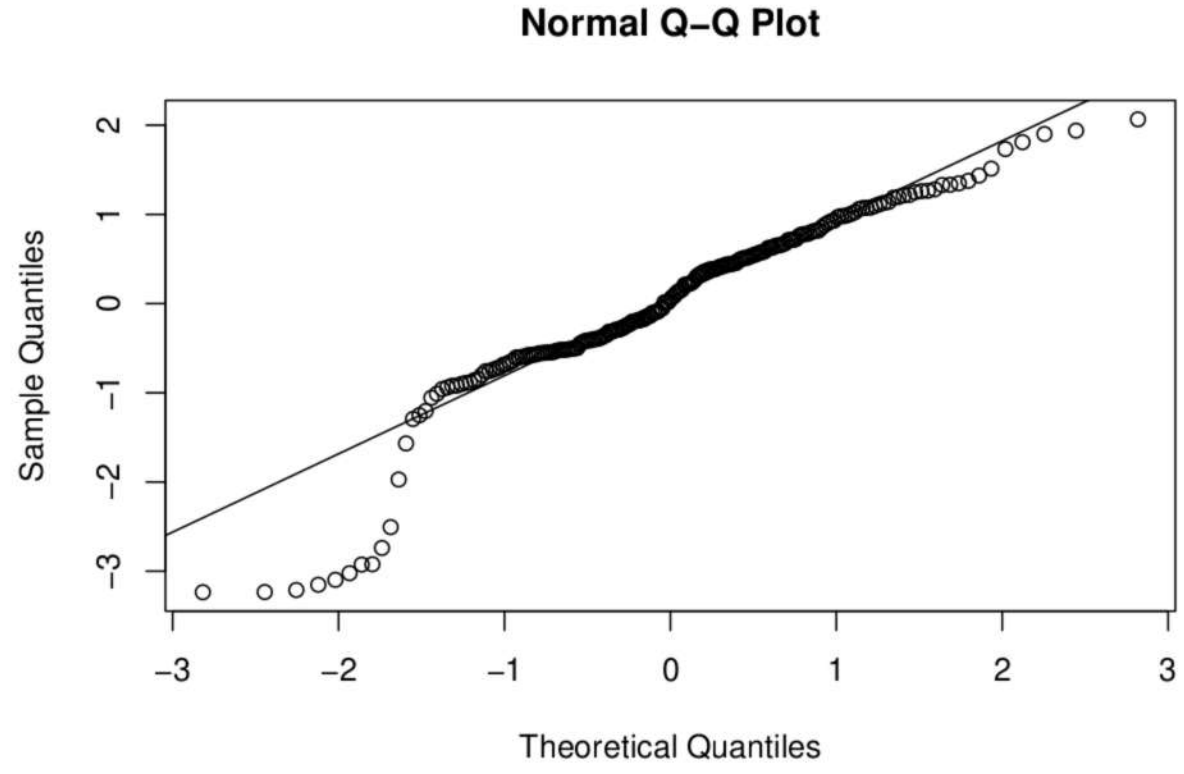
```
summary(modele.mci.after$residuals)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0399544 -0.0064520  0.0006012  0.0000000  0.0081591  0.0255135
```

```
hist(modele.mci.after$residuals, freq=F, col='lightblue')
```



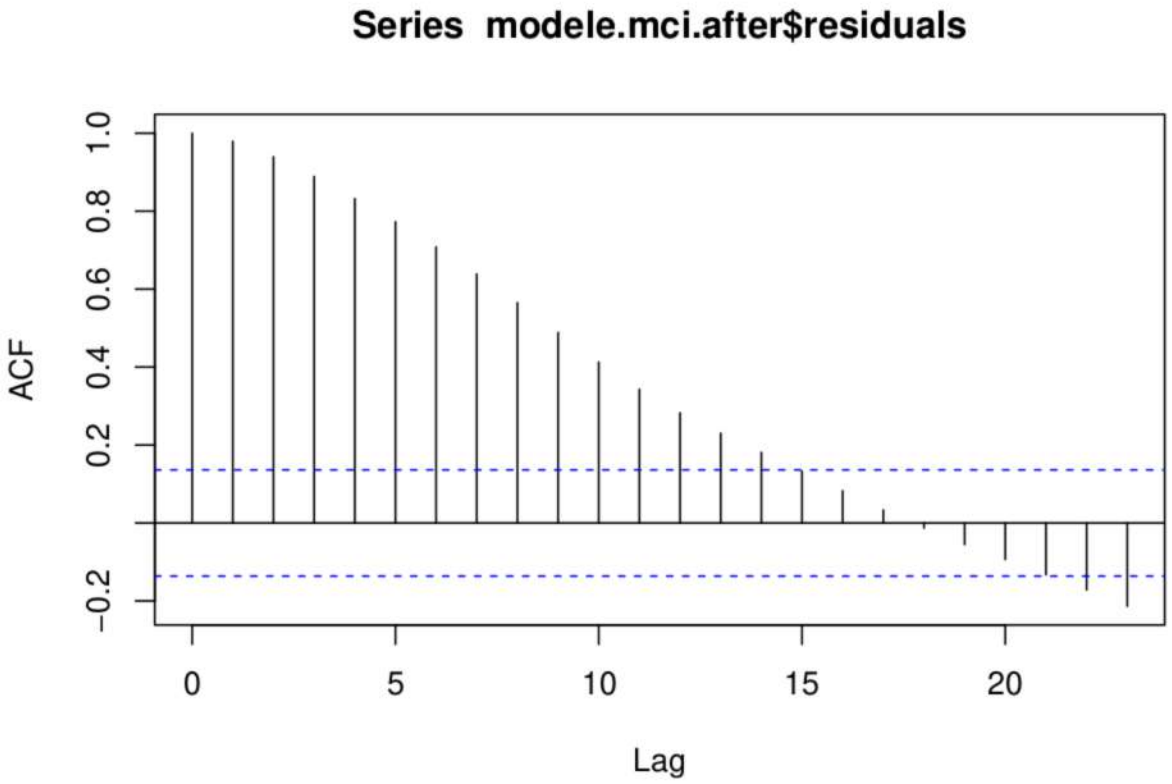

```r
# Autre méthode pour vérifier la normalité des résidus :
stand_res.mci.after = (modele.mci.after$residuals
                       -mean(modele.mci.after$residuals))/sd(modele.mci.after$residuals)
qqnorm(stand_res.mci.after)
qqline(stand_res.mci.after)
```



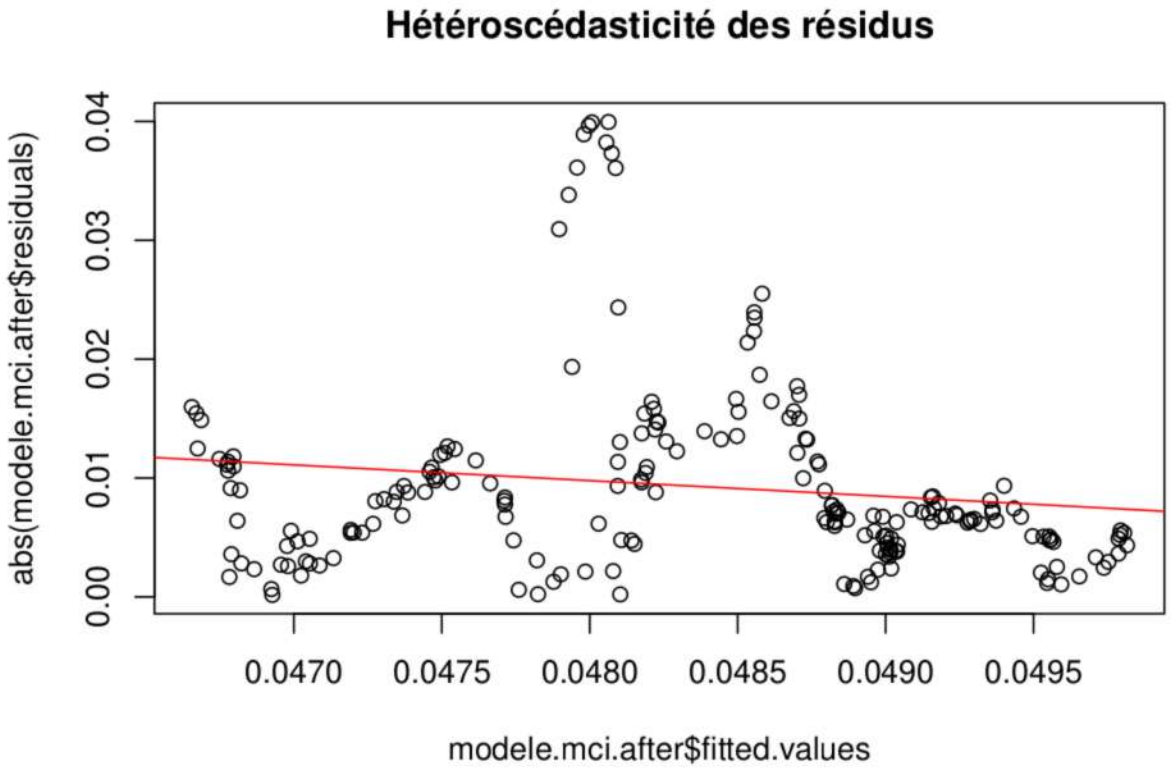
```r
# Finalement on peut faire un test de normalité :
ks.test(modele.mci.after$residuals,pnorm,
        mean=mean(modele.mci.after$residuals),
        sd =sd(modele.mci.after$residuals))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  modele.mci.after$residuals
## D = 0.099152, p-value = 0.03415
## alternative hypothesis: two-sided
```

```r
# Tracé de l'autocorrélation :
acf(modele.mci.after$residuals)
```




```r
# Hétéroscédasticité
plot(abs(modele.mci.after$residuals) ~ modele.mci.after$fitted.values)
r2 <- lm( abs(modele.mci.after$residuals) ~ modele.mci.after$fitted.values )
abline(r2, col="red")
title(main="Hétéroscédasticité des résidus")
```



Notons que l'hétéroscédasticité est nettement améliorée même si ce n'était pas là une priorité du modèle.

Pour conclure sur cette partie, le modèle corrigé a bien permis d'améliorer le pouvoir explicatif d'une nouvelle variable "taux réel". En revanche comme vu précédemment il ne convient pas aux données d'avant 2002 et pour le modèle "global"

PARTIE 3 : Estimation d'une nouvelle spécification: modèle ARMA(p,q)

Question 8 (modèle ARMA(p,q))

Le modèle autorégressif à moyenne mobile (ARMA) est un modèle paramétrique (de paramètres souvent notés p et q) de séries chronologiques stationnaires à courte mémoire. Une série chronologique X t est appelée série ARMA(p,q) si elle vérifie l'équation suivante

$$X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

où: $-\forall k \in [1,n], \phi_k \in \mathbf{C}$ $-\forall k \in [1,n], \theta_k \in \mathbf{C}$ $-\epsilon_t$ est une suite de variables aléatoires complexes, décorrelées, de moyenne nulle et de même variance σ_ϵ^2 On peut définir les polynômes:

$$\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p \theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q$$

En notant B l'opérateur retard, l'équation d'un modele ARMA(p,q) se réécrit alors:

$$\phi(B) X_t = \theta(B) \epsilon_t$$

Question 9 (ordre du modèle)

Pour trouver l'ordre p et q d'un modèle ARMA(p,q) on procède par la minimisation d'une fonction coût à deux variables. Pour ce faire on calcule pour pour tous les couples (p,q) possibles sachant $p < P$ et $q < Q$ (les bornes s'obtiennent en regardant l'AFC: précisemment à partir de quel ordre les valeurs de fonction sont dans l'intervalle autour de 0). Nous travaillons avec des séries ARIMA(p,d,q) où $d = 1$. Nous souhaitons en effet éliminer la tendance de notre série chronologique. Pour la suite on fixe les bornes $P = 4, Q = 4$

```
earning_yield.ts <- ts(data['earning yield'],
                       frequency = 12, start = c(1960,01))
earning_yield.pars <- expand.grid(ar = 1:4, diff = 0, ma = 1:4)
earning_yield.aic <- rep(0, nrow(earning_yield.pars))
earning_yield.bic <- rep(0, nrow(earning_yield.pars))

for (i in seq(along = earning_yield.bic)) {
    out <- arima(earning_yield.ts, unlist(earning_yield.pars[i, 1:3]),
                                        xreg=data$rates)

    earning_yield.aic[i] <- AIC(out)
    earning_yield.bic[i] <- BIC(out)
}
```

```
## Warning in arima(earning_yield.ts, unlist(earning_yield.pars[i, 1:3]), xreg =
## data$rates): possible convergence problem: optim gave code = 1
```

```
i_aic <- which.min(earning_yield.aic)
i_bic <- which.min(earning_yield.bic)
```

On conservera la minimisation par critère BIC qui donne $p = 2, q = 1$. Notre modèle est donc le suivant :

$$X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} = \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

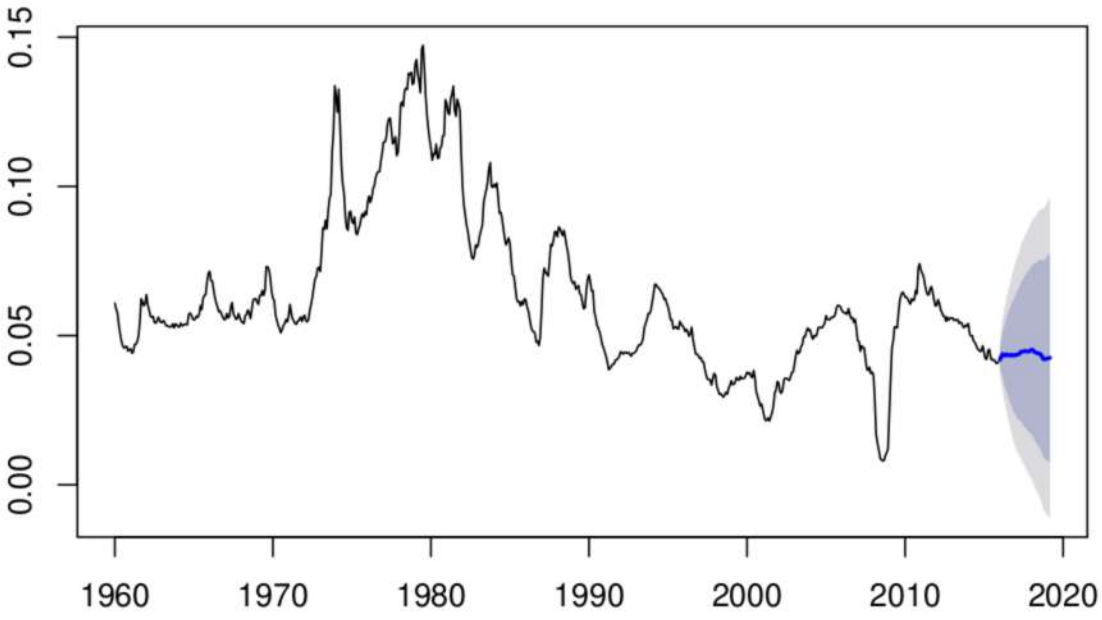
Question 10 (estimation du modèle)

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

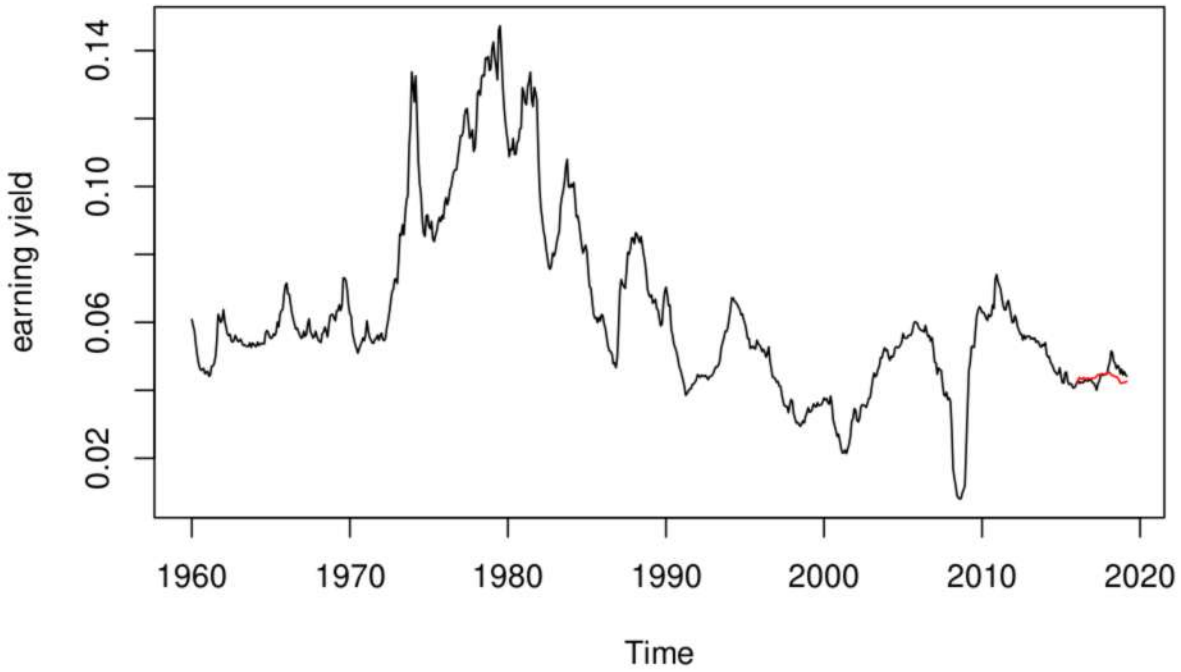
```
# Data
#x_train = window(data[,'rates'], 1960, c(2016,02))
y_train = window(earning_yield.ts, 1960, c(2015,12))
#x_test = window(data[,'rates'], c(2016,3), c(2019,03))
y_test = window(earning_yield.ts, c(2016,01), c(2019,03))

# Modele
fit_ARIMA <- auto.arima(y_train, xreg=data[1:672,'rates'])
forecast_ARIMA <-forecast(fit_ARIMA, xreg=data[673:711,'rates'])
plot(forecast_ARIMA)
```



Question 11 (prévision)

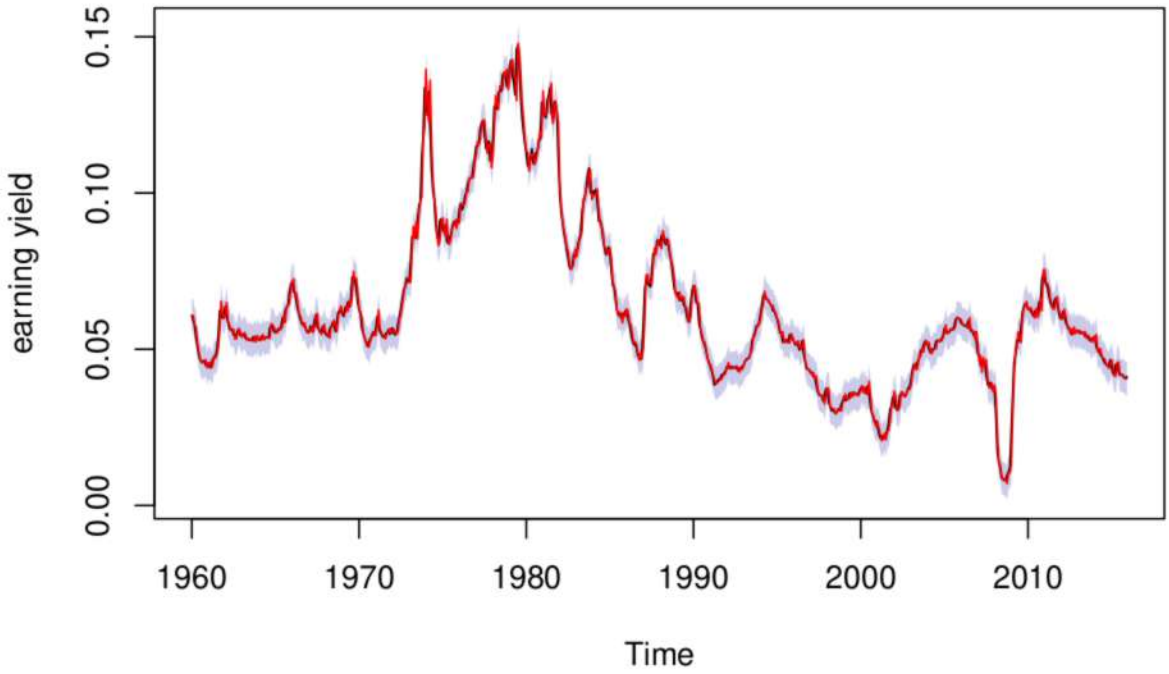
```
plot(earning_yield.ts, type="n")
lines(earning_yield.ts)
lines(forecast_ARIMA$mean,col='red')
```



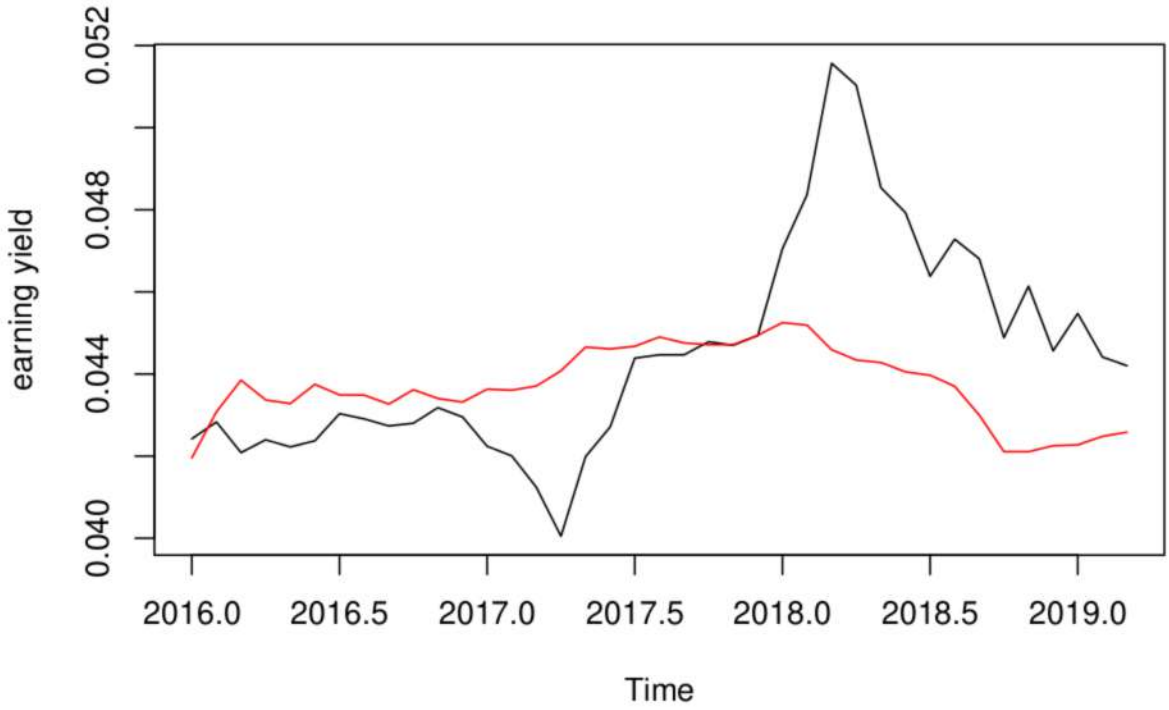

```
upper_train_ARIMA <- fitted(fit_ARIMA) + 1.96*sqrt(fit_ARIMA$sigma2)
lower_train_ARIMA <- fitted(fit_ARIMA) - 1.96*sqrt(fit_ARIMA$sigma2)
upper_test_ARIMA <- forecast_ARIMA$upper[,2]
```

```
lower_test_ARIMA <- forecast_ARIMA$lower[,2]
```

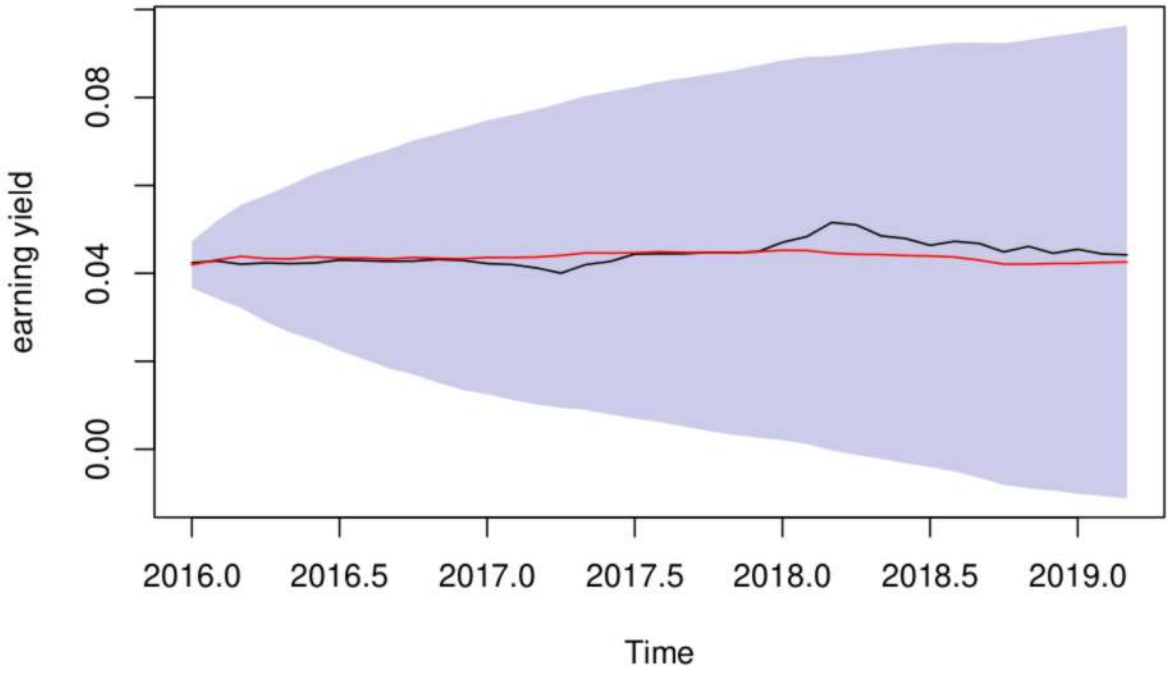
```
plot(y_train, type="n", ylim=range(lower_train_ARIMA,upper_train_ARIMA))
polygon(c(time(y_train),rev(time(y_train))), c(upper_train_ARIMA,rev(lower_train_ARIMA)),
        col=rgb(0,0,0.6,0.2), border=FALSE)
lines(y_train)
lines(fitted(fit_ARIMA), col='red')
```




```
plot(y_test, type="n")
lines(y_test)
lines(forecast_ARIMA$mean,col='red')
```




```
plot(y_test, type="n", ylim=range(upper_test_ARIMA,lower_test_ARIMA))
polygon(c(time(y_test),rev(time(y_test))), c(upper_test_ARIMA,rev(lower_test_ARIMA)),
        col=rgb(0,0,0.6,0.2), border=FALSE)
lines(y_test)
lines(forecast_ARIMA$mean,col='red')
```



Le premier tracé montre la série complète et la prédiction sur l'horizon de 3 périodes (de 2016 à 2019). Le deuxième graphique trace la série et son interval de confiance à 95%. Ce qui nous intéresse -la prédiction- se trouve sur le 3ème graphique. Sur le 4ème graphique on y ajoute l'intervalle de confiance à 95%. On peut affirmer que notre modèle est plutôt bon pour la plage de prédiction donnée.

Question 12 (comparaison de critères)

Pour sélectionner un modèle sur sa capacité à prédire, on recours à deux critères : le Root Mean Square Error (RMSE) et la Mean Absolute Error (MAE) définis comme suit :

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \; RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Premier modèle estimé:

```
predictions_ARIMA = forecast_ARIMA$mean
# Root Mean Square Error
RMSE <- function(error) { sqrt(mean(error^2)) }
RMSE(predictions_ARIMA - y_test)
```

```
## [1] 0.002611064
```

```
# Function for Mean Absolute Error
MAE <- function(error) { mean(abs(error)) }
MAE(predictions_ARIMA - y_test)
```

```
## [1] 0.001960378
```

2ème modèle estimé:

```
fit_ARMA<- arima(y_train, unlist(earning_yield.pars[2, 1:3]),
                 xreg=data[1:672,'rates'])
predictions_ARMA = predict(fit_ARMA, newxreg=data[673:711,'rates'])$pred
RMSE(predictions_ARMA - y_test)
```

```
## [1] 0.002785972
```

```
MAE(predictions_ARMA - y_test)
```

```
## [1] 0.00201094
```

Au vue des critères précédents il semble que ce soit la 1ère méthode qui soit la plus performante.

Question 11 bis (Test de Diebold et Mariano)

On utilise la fonction dm.test fourni par R qui permet de comparer la performance prédictive de deux méthodes. Ici on rejette l'hypothèse nulle: "la méthode 2 est moins prédictive que la méthode 1".

```
dm.test(e1=fit_ARMA$residuals,e2=forecast_ARIMA$residuals,
        alternative='less',h=1)
```

```
##
##  Diebold-Mariano Test
##
## data:  fit_ARMA$residualsforecast_ARIMA$residuals
## DM = 0.27176, Forecast horizon = 1, Loss function power = 2, p-value =
## 0.6071
## alternative hypothesis: less
```

PARTIE 4: Stabilité du modèle

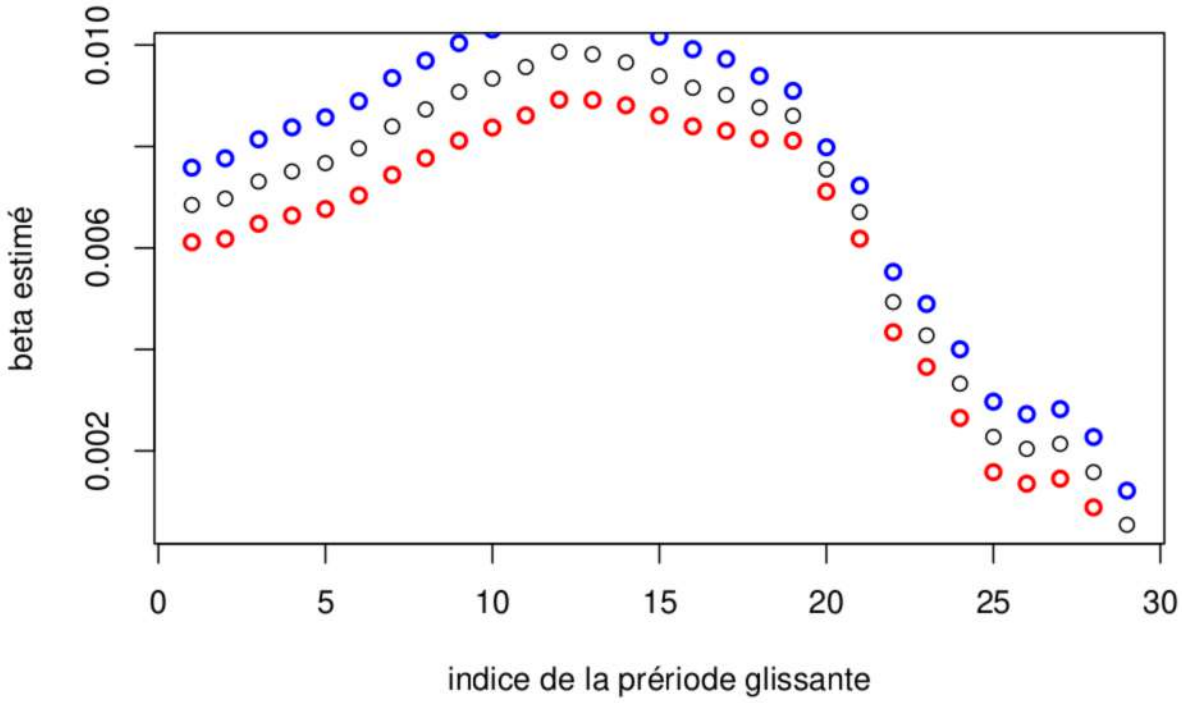
Dans cette partie on s'intéresse à la stabilité du coefficient β dans l'équation du Fed model :

$$\frac{E_t}{P_t} = \alpha + \beta r_t$$

Question 13 (méthode glissante)

```
begin = 1960
end = 1990
beta = rep(0, 2019-end)
beta_inf = rep(0, 2019-end)
beta_sup = rep(0, 2019-end)
earning_yield <- ts(data=data$'earning yield', start=c(1960,1), frequency=12)
rates <- ts(data=data$'rates', start=c(1960,1), frequency=12)
for (i in seq(along = beta)) {
  y = window(earning_yield,  start=begin+i, end=end + i)
  x = window(rates, start=begin+i, end=end + i)
  reg <-lm(y ~ x)
  conf = confint(reg, level=0.95)
  beta_inf[i] <- conf[2,1]
  beta_sup[i] <- conf[2,2]
  beta[i] <- reg$coefficients[2]
}

plot(beta, xlab='indice de la prériode glissante',ylab='beta estimé', main='')
points(beta_inf, lwd=2, col="red")
points(beta_sup, lwd=2, col="blue")
```



Dans cette première question, on cherche à estimer le coefficient β selon la méthode glissante. On observe donc

que l'estimation de β ne semble pas rester constante en fonction des fenêtres utilisées, ainsi des changements de structure semblent être mis en évidence. Afin d'être sur qu'il ne s'agit pas simplement de bruit, nous rajoutons les intervalles de confiance à 95%. On note alors que les bornes basses de l'intervalle de confiance de l'estimation de β pour les premières fenêtres sont largement au-dessus des bornes hautes de l'intervalle de confiance de l'estimation de β pour les dernières fenêtres.

Question 14 (Méthode CUSUM)

Le modèle CUSUM est basé sur une estimation récursive des coefficients de l'équation. On calcule les résidus des estimations récursives. Sous l'hypothèse nulle (c'est à dire pas de changement structurel), ces estimations tendent vers un processus de Wiener.
On doit donc rejeter l'hypothèse nulle (absence de changement structurel) si le processus observé empiriquement s'éloigne de manière trop importante des valeurs d'un processus de Wiener. Dans notre cas, les fluctuations observées excèdent par endroit fortement les zones de confiance à 95%; on doit donc rejeter l'hypothèse nulle. Il y a donc un changement structurel ce qui rejoint la conclusion de la question précédente.

```
library(strucchange)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

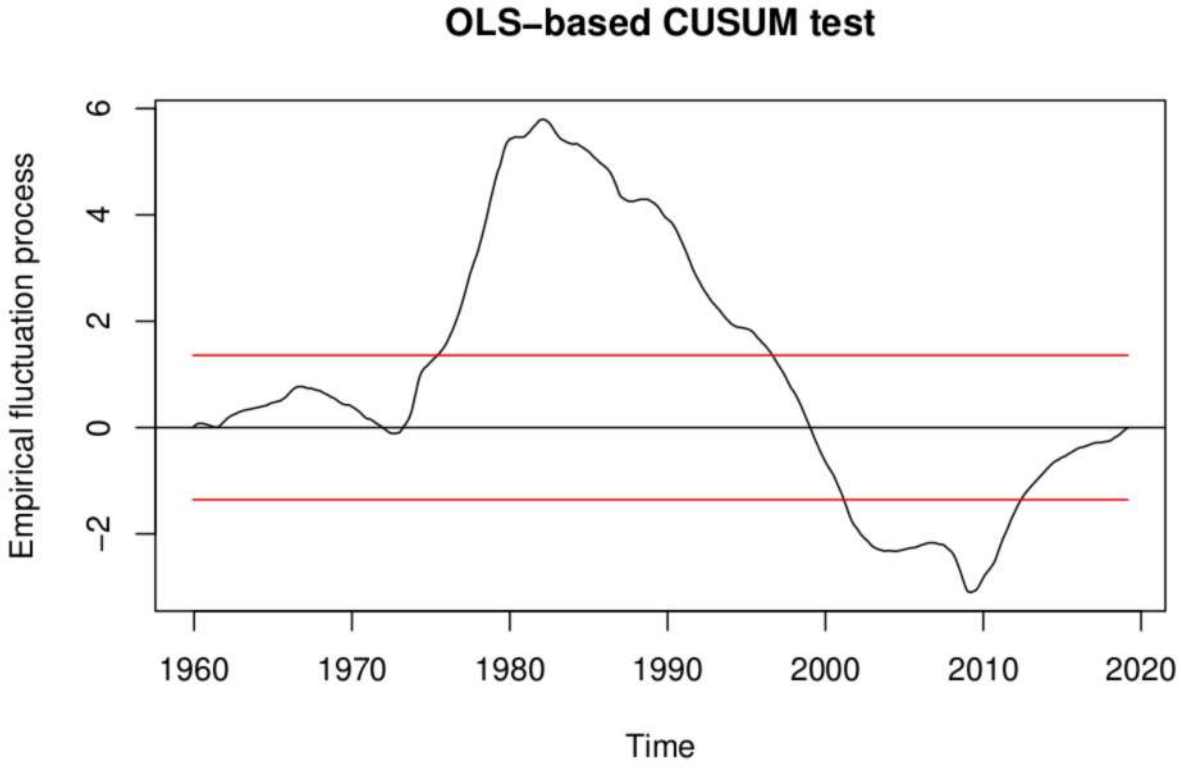
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
##
## Attaching package: 'strucchange'
```

```
## The following object is masked from 'package:stringr':
##
##     boundary
```

```
l <- formula(lm(earning_yield ~ rates))
ocus <- efp(l, type="OLS-CUSUM")
plot(ocus)
```


OLS−based CUSUM test
Empirical fluctuation process
6
4
2
0
−2
1960
1970
1980
1990
2000
2010
2020
Time