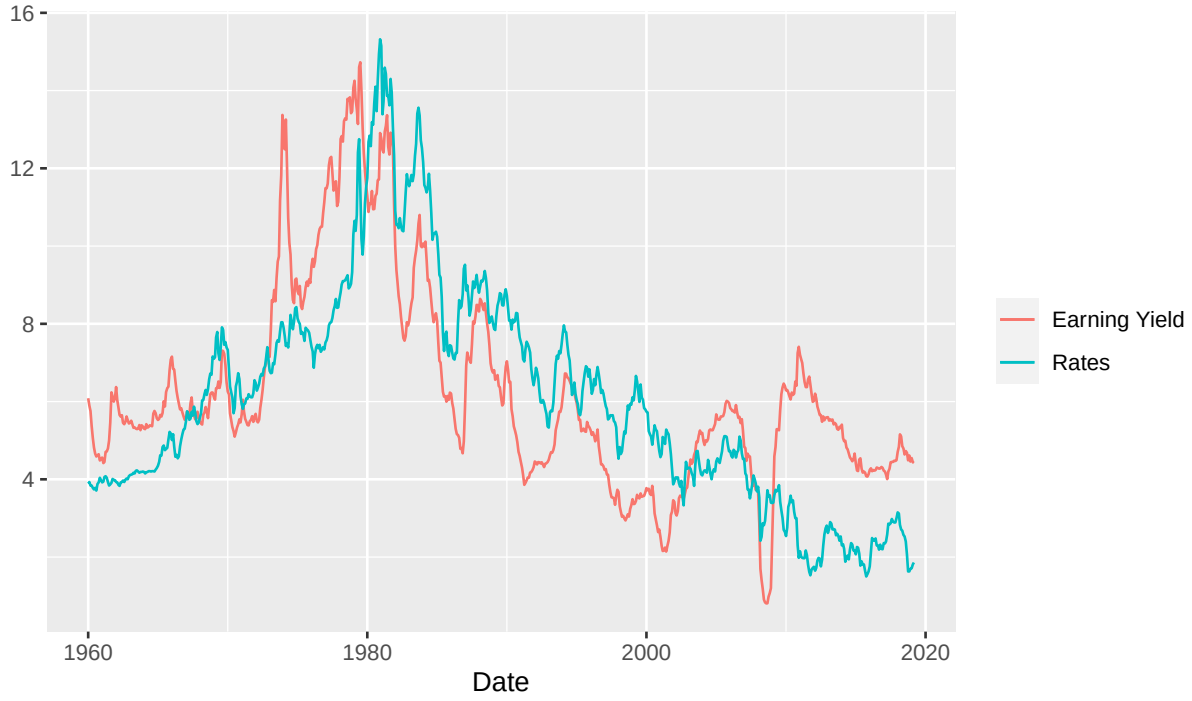
HACHICHA - CENTRALE

On commence par la lecture des données, l'ajout de la colonne dates et le calcul du Earnings Yield

```
df = read.csv(file='061120_data.csv', sep=';', header=T)
df['date'] = seq(from = as.Date("1960-01-01"), to = as.Date("2019-03-01"), by = 'month')
df['ey'] = 100 * (df$earnings / df$price)
```

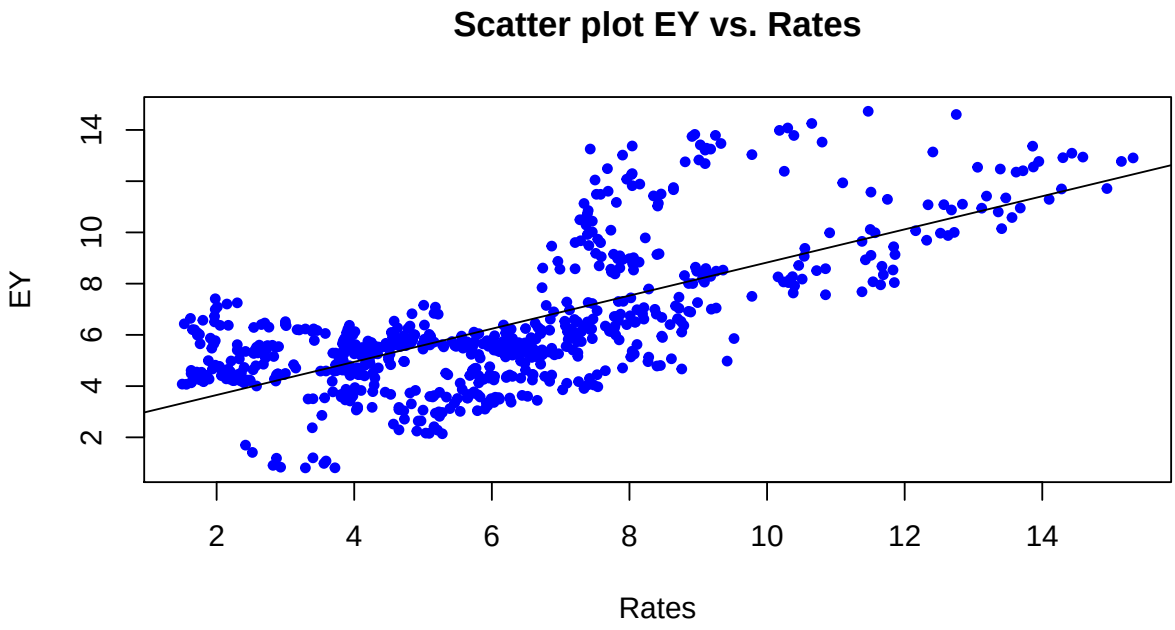
Question 1

```
library(ggplot2)
ggplot(data = df) +
  geom_line(aes(x = date, y = ey, color = "Earning Yield")) +
  geom_line(aes(x = date, y = rates, color = "Rates")) +
  xlab('Date') +
  ylab('') +
  labs(color='')
```



On remarque que les deux courbes suivent généralement la même tendance. On vérifie la relation linéaire en faisant le plot suivant.

```r
model = lm(ey~rates, data = df)
plot(df$rates, df$ey, xlab = 'Rates', ylab = 'EY', pch=20, col='blue', title('Scatter plot EY vs. Rates
abline(model)
```



On constate donc une certaine relation affine entre les deux variables.

Question 2

La méthode des moindres carrés ordinaires, ou MCO, a pour objectif, étant donné des observations d'une variable $(y_i)_{i\in[1,..n]}$ et des observations de variables explicatives $(X_i)_{i\in[1,..n]}$, de lier ces observations par la relation: $y = X\beta + \epsilon$ avec β un vecteur coefficients ou des poids pour pondérer les X_i et ϵ un vecteur d'erreur qui suit la loi normale centrée $N(0, \sigma^2 I)$.

Une estimation de β est $\hat{\beta} = \underset{\beta}{argmin}\ \|y - X\beta\|^2$

Les hypothèses du modèle linéaire sont: - Les varaibles explicatives ne sont pas corrélées. - Les ϵ_i sont décorrélées. Ceci est exigé déjà par l'hypothèse $\epsilon \sim N(0, \sigma^2 I)$ - Homoscédasticité: Les (ϵ_i) ont une variance constante σ^2 - Exogénéité: Les variables explicatives ne sont pas corrélées aux termes d'erreurs.

Question 3

On définit la fonction d'erreur : $f(\alpha, \beta) = \sum_{t=0}^{n} (\frac{E_t}{P_t} - (\alpha + \beta r_t))^2$

Pour minimiser f, on écrit: $\frac{\partial f}{\partial \alpha} = 0$

librarlibr$\Leftrightarrow n.\alpha + \sum_{t=0}^{n} (\beta r_t - \frac{E_t}{P_t}) = 0$

donc $\hat{\alpha} = (\frac{\bar{E}}{P}) - \hat{\beta}\bar{r}$

Et aussi nous avons que:

$\frac{\partial f}{\partial \beta} = 0 \Leftrightarrow \sum_t r_t((\frac{E}{P})_t - \hat{\alpha} - \hat{\beta} r_t) = 0 \Leftrightarrow \hat{\beta} = \frac{\sum_t (r_t - \bar{r})((\frac{E}{P})_t - (\frac{\bar{E}}{P}))}{\sum_t (r_t - \bar{r})^2} = \frac{Cov(\frac{E}{P}, r)}{\sigma_r^2}$

$Var(\hat{\beta}) = Var(\frac{\sum_t (r_t - \bar{r}).(\frac{E}{P})_t}{\sum_t (r_t - \bar{r})^2}) = Var(\frac{\sum_t (r_t - \bar{r}).(\beta r_t + \alpha + \epsilon_t)}{\sum_t (r_t - \bar{r})^2}) = Var(\frac{\sum_t (r_t - \bar{r}).\epsilon_t}{\sum_t (r_t - \bar{r})^2})$ car la seule variable aléatoire ici est ϵ_t.

Ainsi: $Var(\hat{\beta}) = \frac{\sigma^2 . \sum_t (r_t - \bar{r})^2}{(\sum_t (r_t - \bar{r})^2)^2} = \frac{\sigma^2}{\sum_t (r_t - \bar{r})^2} = \frac{\sigma^2}{T.\sigma_r^2}$

Avec σ_r l'écart type des taux sans risques. D'où en supposant que σ_r reste constant (ou au moins borné) quand $T \to +\infty$, nous avons $Var(\hat{\beta}) \overset{T \to +\infty}{\to} 0$

Question 4

```
summary(model)
```

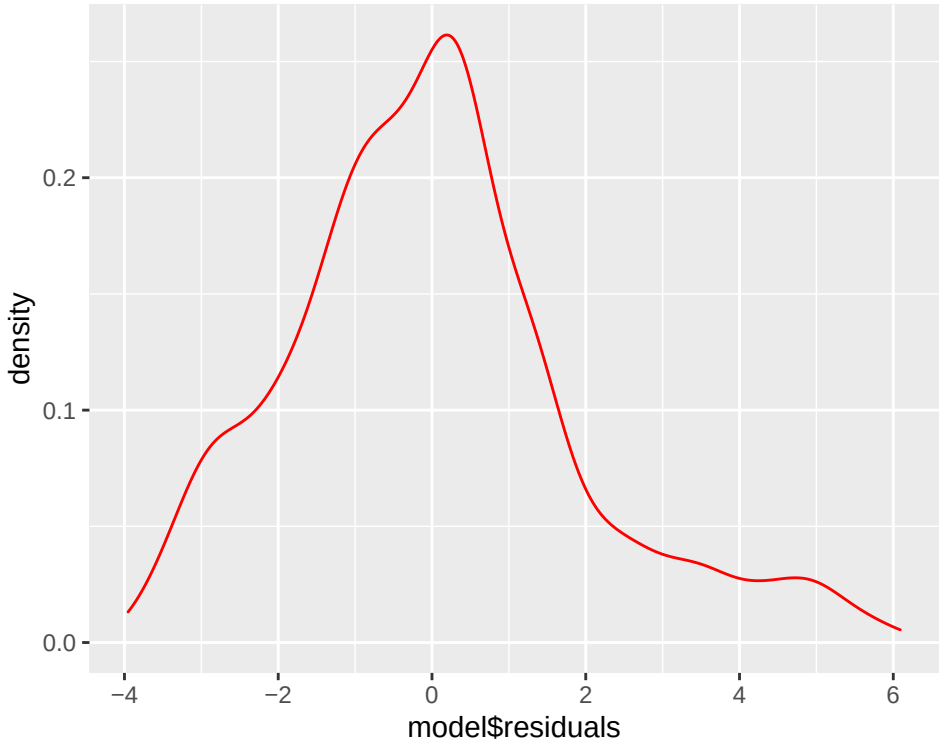
```
##
## Call:
## lm(formula = ey ~ rates, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9533 -1.2084 -0.0824  0.9177  6.0914
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.35860    0.16389   14.39   <2e-16 ***
## rates        0.64668    0.02438   26.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.875 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
```

Les p-values du test de student pour l'intercept et pour la variables *rates* sont très faibles, ce qui signifie que les termes d'intercept de rates sont significatifs dans le modèle. La p-value du test de Fisher (en bas) est très faible aussi. Ceci affirme donc que le modèle linéaire est significatif.

Cependant, on a un $R^2_{adjusted} = 0.4974$ qui est assez faible, ce qui signifie que la qualité d'ajustement du modèle n'est pas très bonne.

Question 5

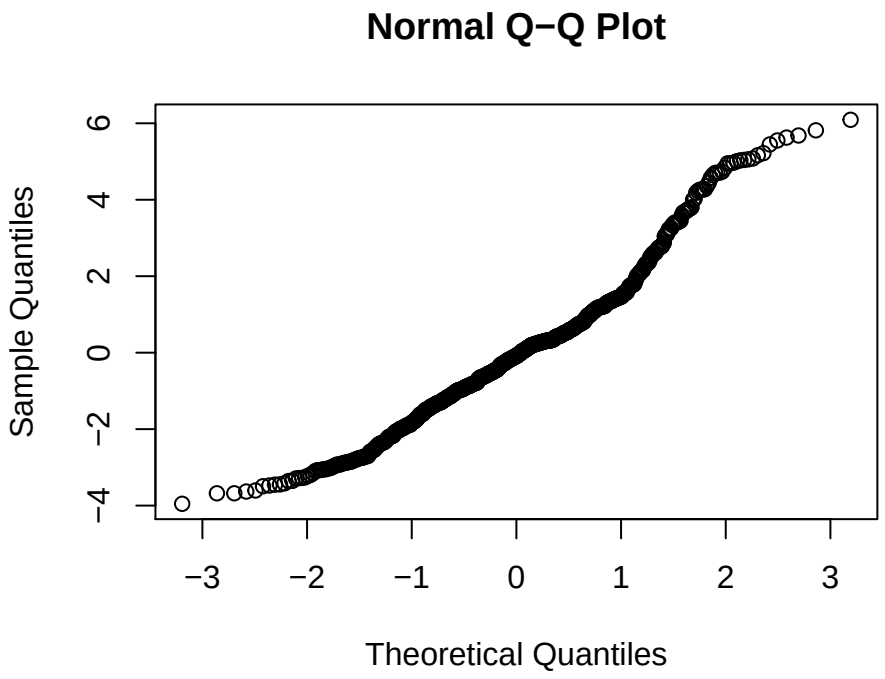
```
ggplot(data=model, aes(x=model$residuals)) +
  geom_density(color="red")
```



Normalité des résidus :

La densité tracée n'est pas très similaire à une densité normale. Il y a beaucoup d'irrégularités, une asymétrie et une queue (distribution tail) assez longue à droite.

```
qqnorm(model$residuals)
```



On voit calirement que le qqplot n'est pas une ligne droite parfaite. Ceci confirme que la distribution des résidus est 'skewed' (biaisée en français?)

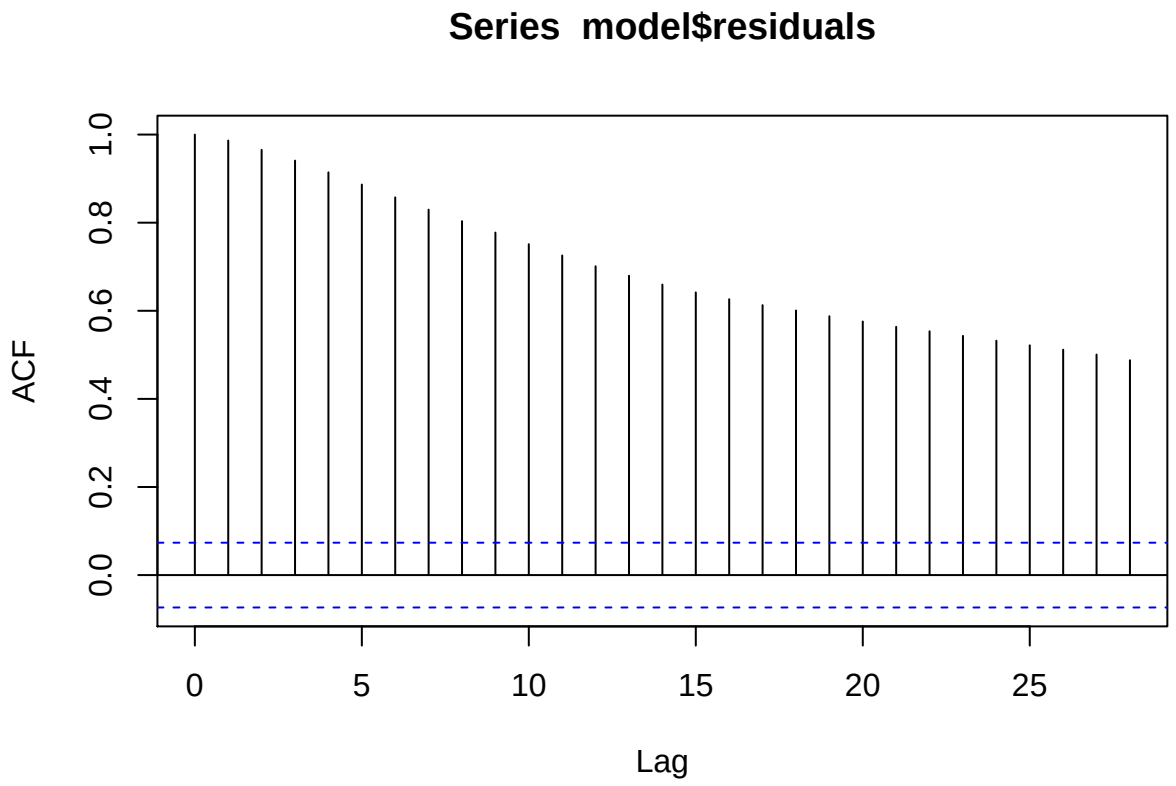
Autocorrélation des résidus :

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

```
acf(model$residuals)
```

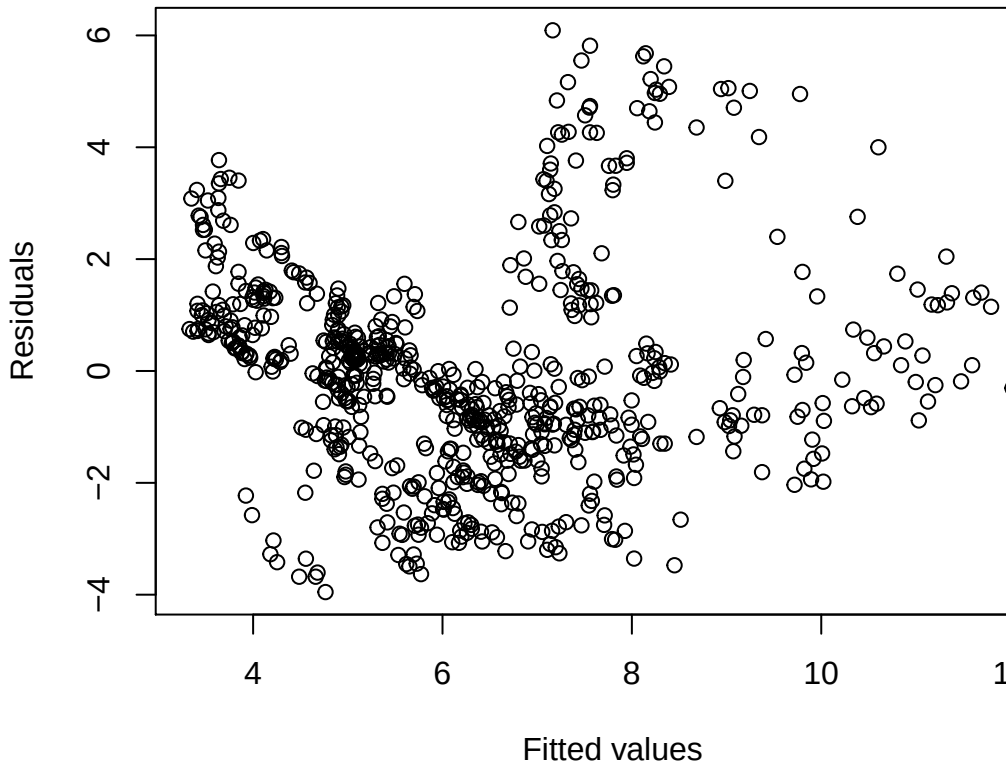



```
bgtest(model)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  model
## LM test = 692.18, df = 1, p-value < 2.2e-16
```

Le plot de l'ACF et la très faible p-value du test de Breusch-Godfrey montrent clairement qu'il y a une autocorrélation entre les résidus du modèle linéaire.

```r
plot(model$fitted.values, model$residuals, xlab = 'Fitted values', ylab='Residuals')
```



Recherche d'hétéroscédasticité

Les résidues ne semblent pas centrés autour de la ligne 0 et ils ne sont pas régulièrement répartis autour de cette ligne ce qui suggère que leur variance n'est pas constante (la variance augmente notamment quand les fitted values augmentent)

```r
bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 19.049, df = 1, p-value = 1.274e-05
```

On a une très faible p-value donc on rejette l'hypothèse nulle d'homoscedasticité. Le test de Breush-Pagan confirme l'existence d'hétéroscédasticité.

Pour la traiter on peut utiliser un GLS (Generalized Least Squares). Ce modèle essaie de minimiser $(y - X\beta)^T \Sigma^{-1} (y - X\beta)$ où Σ est la matrice de covariance des résidus de notre premier modèle MCO.

Dans ce cas, puisque Σ est symétrique, on peut écrire $\Sigma = A^T A$ et on obtient:

$(y - X\beta)^T \Sigma^{-1} (y - X\beta) = (A^{-1}y - A^{-1}X\beta)^T (A^{-1}y - A^{-1}X\beta) = \left\| A^{-1}y - A^{-1}X\beta \right\|$

D'où le GLS revient à faire une OLS $y' = X'\beta + \epsilon'$ où on $y' = A^{-1}y$, $X' = A^{-1}X$

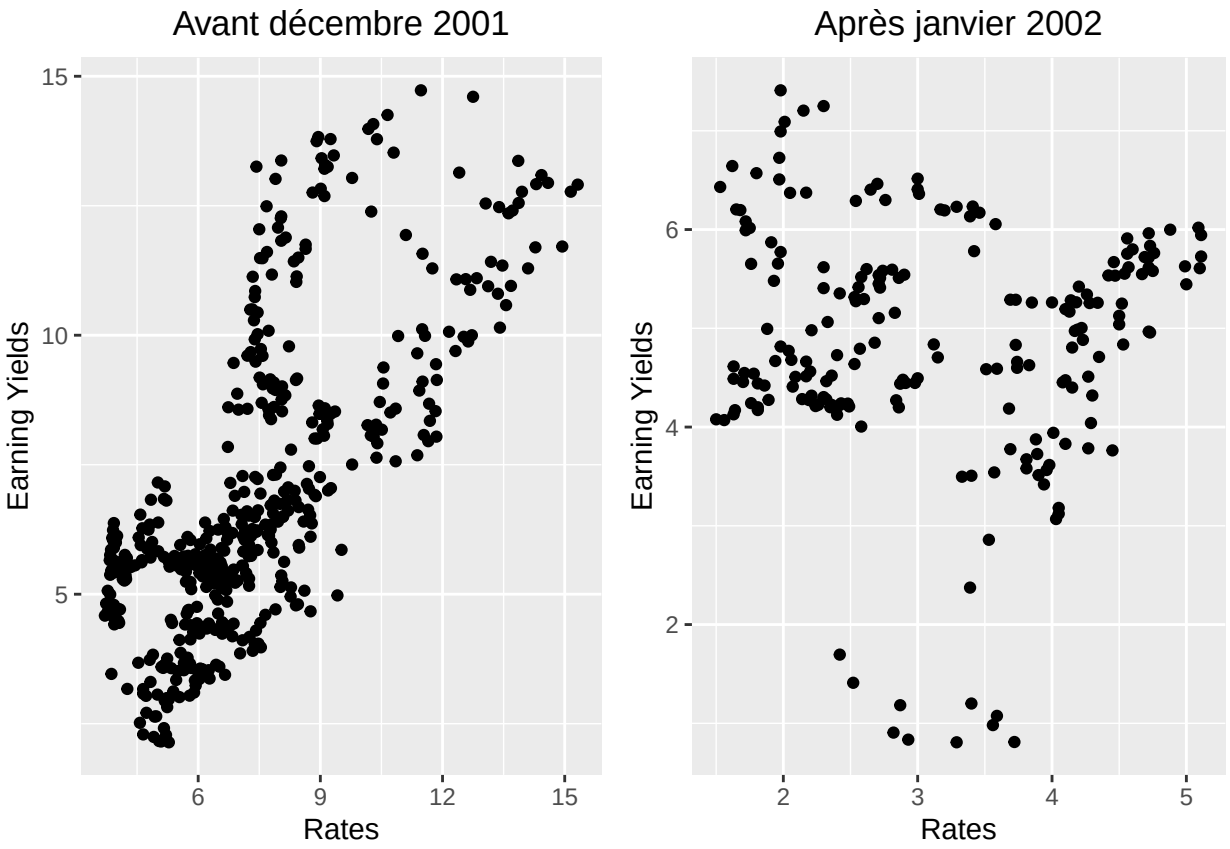
Question 6

On observe à partir du plot des deux séries temporelles de rates et earning yield qu'à partir de l'an 2000 à peu près on quelques perturbations qui font que les deux séries ne suivent pas forcément les mêmes tendances.

On peut utiliser le test de Chow pour déterminer la date de rupture la plus probable. On décide de découper le dataset en deux parties : avant et après janvier 2002.

```
library(gridExtra)

df_1 = df[df$date <= '2001-12-31', ]
df_2 = df[df$date > '2001-12-31', ]
plot_1 = ggplot(data = df_1) +
  geom_point(aes(x = rates, y = ey)) +
  xlab('Rates') +
  ylab('Earning Yields') +
  labs(title = 'Avant décembre 2001') +
  theme(plot.title = element_text(hjust = 0.5))
plot_2 = ggplot(data = df_2) +
  geom_point(aes(x = rates, y = ey)) +
  xlab('Rates') +
  ylab('Earning Yields') +
  labs(title='Après janvier 2002') +
  theme(plot.title = element_text(hjust = 0.5))
grid.arrange(plot_1, plot_2, ncol = 2)
```



Pour la période jusqu'à décembre 2001, on observe une certaine linéarité entre les deux variables. Par contre, pour seconde période, on n'arrive pas à identifier visuellement une corrélation entre les taux d'intérêt et les Earning Yields.

On entraine deux modèle de régression linéaire sur les deux périodes respectives:

```
model_1 = lm(ey~rates, data=df_1)
summary(model_1)
```

Période 1:

```
##
## Call:
## lm(formula = ey ~ rates, data = df_1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6507 -1.3412 -0.3798  1.0548  6.2518
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.93908    0.26783   3.506 0.000495 ***
## rates        0.81615    0.03478  23.468  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.958 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
```

Le test de Fisher montre que le modèle est significatif. Les tests de student montrent que les variables sont significatives.

```
model_2 = lm(ey~rates, data=df_2)
summary(model_2)
```

Période 2

```
##
## Call:
## lm(formula = ey ~ rates, data = df_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0174 -0.6078  0.0674  0.8040  2.5248
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.97502    0.27822  17.881   <2e-16 ***
## rates       -0.04553    0.08459  -0.538    0.591
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.24 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
```

Le test de Fisher a une p-value élevée ce qui indique qu'on peut rejeter l'hypothèse de significativité de ce modèle. Pour traiter ce type de configuration de données, on peut recourir aux familles de modèles de régréssions non linéaires, comme par exemple le random forest.

Partie 2: Estimation d'une nouvelle spécification et comparaiso

Question 7

```
library(data.table)
df['real_rates'] =df$rates - 100 * (df$cpi - shift(df$cpi, 12)) / shift(df$cpi, 12)
model_q7 = lm(ey~real_rates, data = df)
summary(model_q7)
```

```
##
## Call:
```

```
## lm(formula = ey ~ real_rates, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2899 -1.7276 -0.7049  0.7556  7.7147
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.6186     0.1403  47.164  < 2e-16 ***
## real_rates   -0.1348     0.0425  -3.171  0.00159 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.645 on 697 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.01422,    Adjusted R-squared:  0.01281
## F-statistic: 10.05 on 1 and 697 DF,  p-value: 0.001587
```

Les tests de student et de Fisher montrent que le modèle ainsi que les variables utilisés sont significatifs.

Néanmoins le R-squared (0.01422) est très bas !

On fait alors une régression linéaire sur chacune des deux périodes :

```
df_1 = df[df$date <= '2001-12-31', ]
model_1 = lm(ey~real_rates, data=df_1)
summary(model_1)
```

Première période

```
##
## Call:
## lm(formula = ey ~ real_rates, data = df_1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6651 -1.6983 -0.9737  1.9155  7.7162
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.79051    0.18826   41.38  < 2e-16 ***
## real_rates  -0.30229    0.04956   -6.10 2.15e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.751 on 490 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.07058,    Adjusted R-squared:  0.06868
## F-statistic: 37.21 on 1 and 490 DF,  p-value: 2.154e-09
```

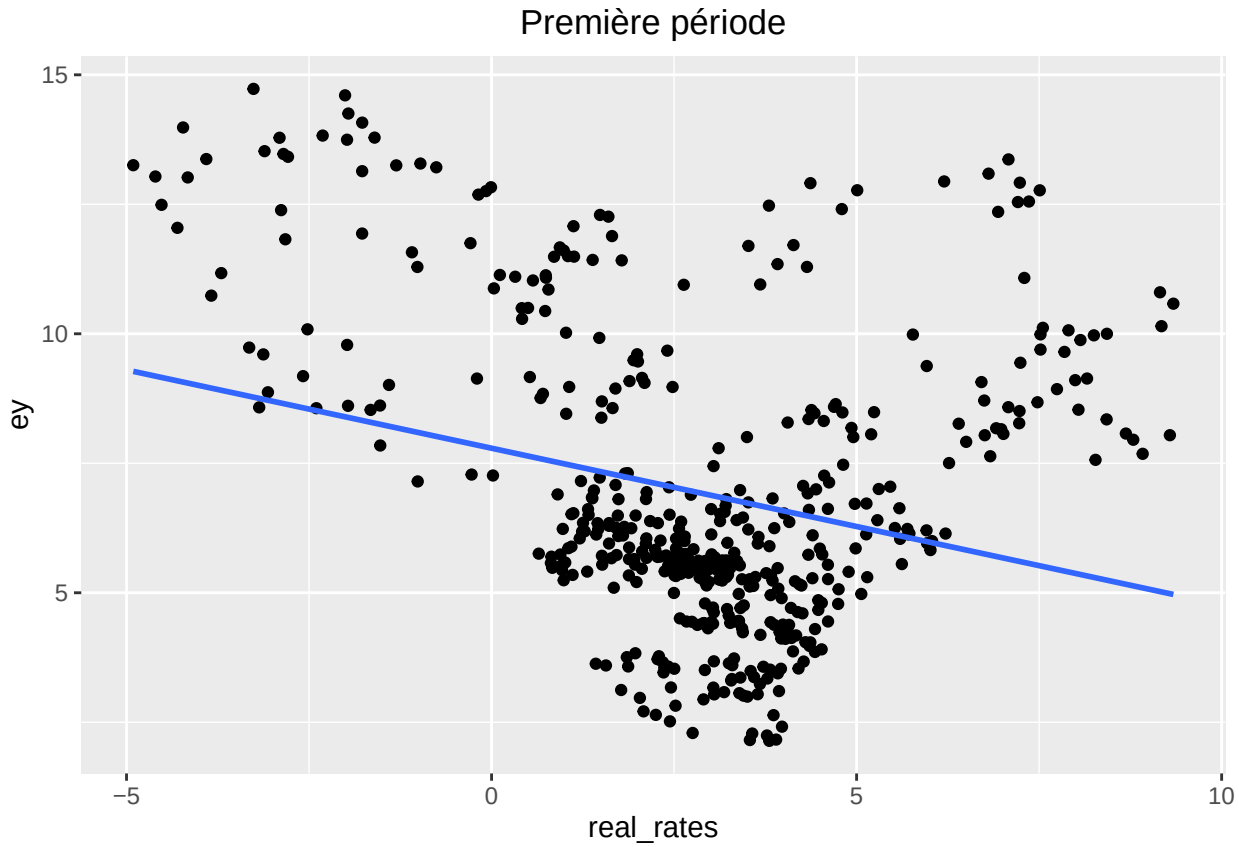
Les tests de student et de Fisher sont concluants. Néanmoins, le R^2 est très faible.

```
library(ggplot2)
ggplot(data = df_1, aes(x = real_rates, y = ey)) +
  geom_point(color='black') +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title='Première période') +
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 12 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 12 rows containing missing values (geom_point).
```



En effet, on n'a plus une tendance affine entre le earning yield et les taux d'intérêt réels. On peut retenir le premier modèle (earning yield en fonction des taux sans risques seulement) pour la première période.

```
df_2 = df[df$date > '2001-12-31', ]
model_2 = lm(ey~real_rates, data=df_2)
summary(model_2)
```

Pour la deuxième période:

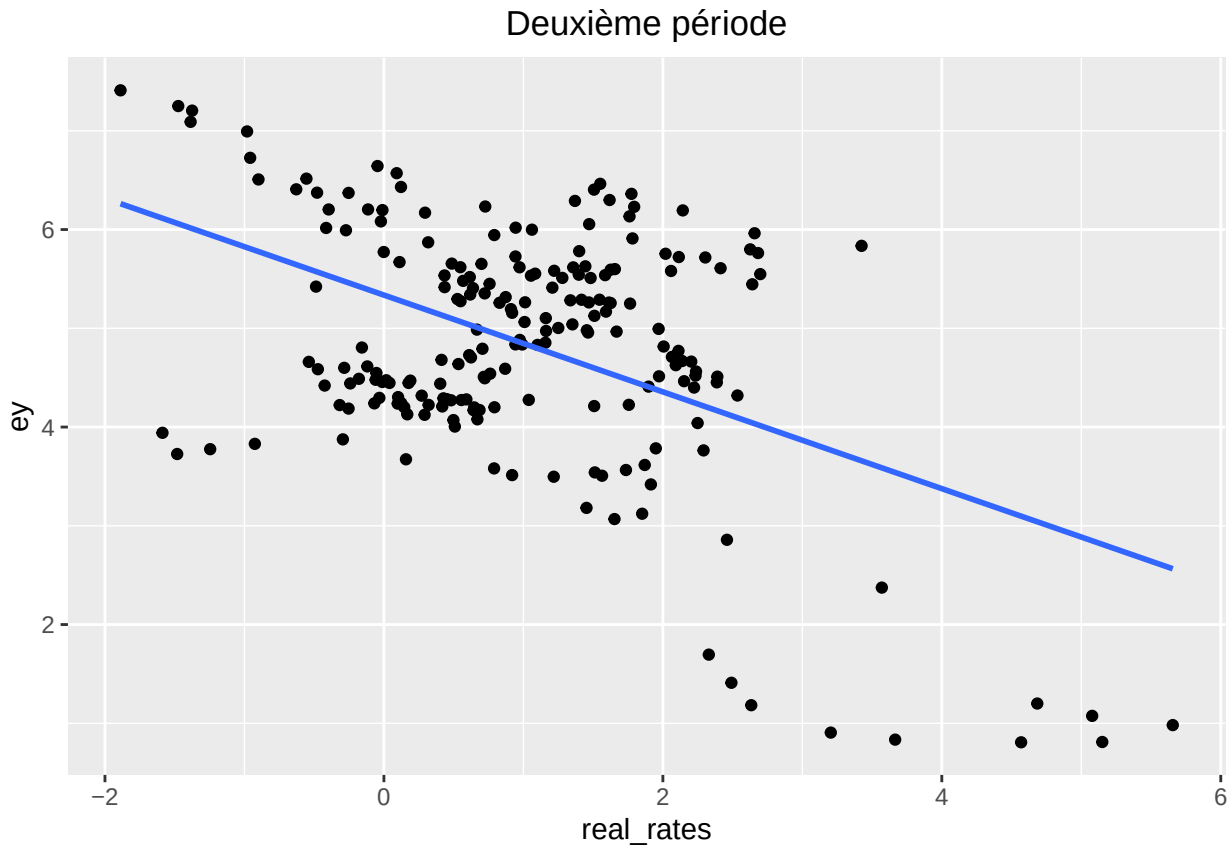
```
##
## Call:
## lm(formula = ey ~ real_rates, data = df_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8622 -0.8730  0.2796  0.7709  2.1766
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.33634    0.09935  53.713  < 2e-16 ***
## real_rates  -0.49000    0.06260  -7.827 2.62e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.089 on 205 degrees of freedom
## Multiple R-squared:  0.2301, Adjusted R-squared:  0.2263
## F-statistic: 61.26 on 1 and 205 DF,  p-value: 2.621e-13
```

On observe pas mal d'améliorations pour la deuxième période avec la prise en compte des taux réels. En effet, le test de Fisher montre que le modèle est significatif et les tests de Student montrent que les variables explicatives sont significatives également

Le $R^2 = 0.2301$ est bas, mais en tout cas il est bien meilleur que celui obtenu avec le premier modèle (avec les taux sans risques) où le R^2 était égal à 0.001411.

```
ggplot(data = df_2, aes(x = real_rates, y = ey)) +
  geom_point(color='black') +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title='Deuxième période') +
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Le scatter plot confirme la colinéarité (même si elle n'est pas très forte) entre le earning yield et les taux d'intérêt réels pour cette deuxième période.

En conclusion:

- Pour la première période : on peut retenir le premier modèle (earning yield en fonction des taux sans risques seulement).
- Pour la deuxième période : on peut retenir le deuxième modèle avec les taux d'intérêt réels.

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

Question 8

Un modèle ARMA(p,q) est un modèle de séries temporelles où, pour un processus temporel X_t on suppose que : $X_t = \epsilon_t + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j}$ où

- Les ϕ_i et les θ_j sont des paramètres du modèle.
- Les ϵ_j sont des termes d'erreurs (bruits blancs)

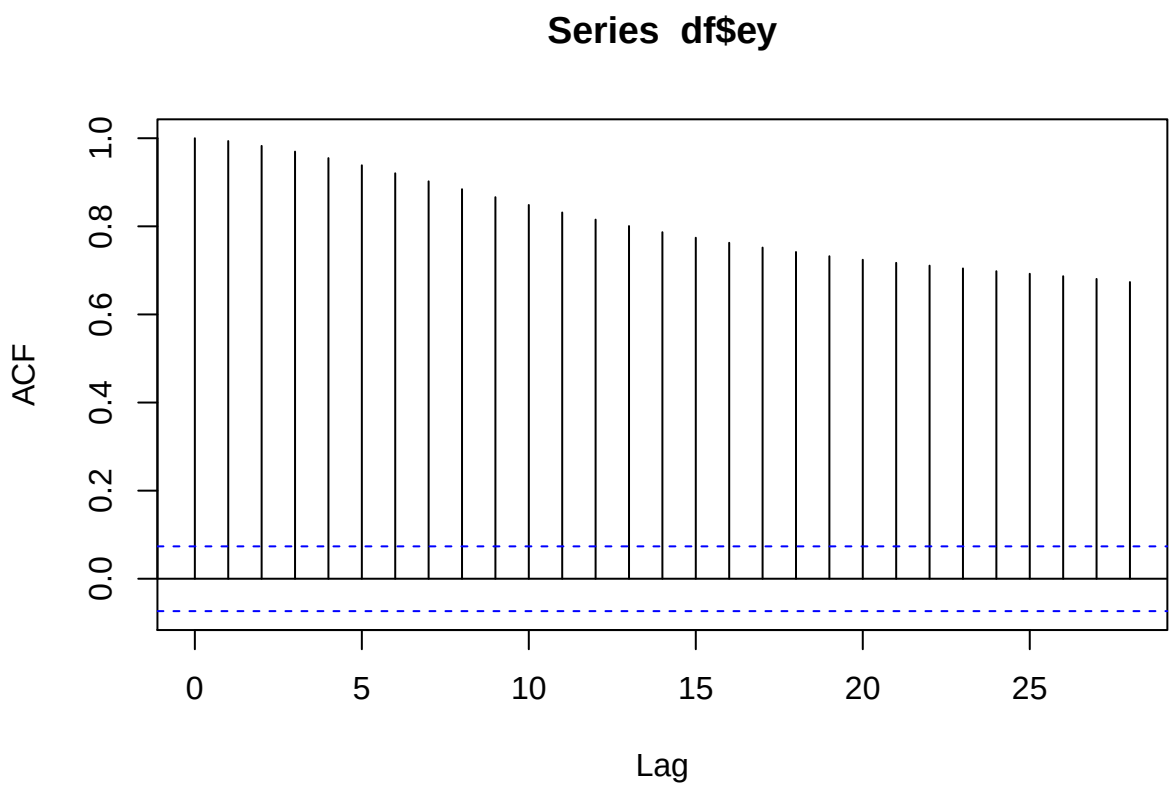
Un modèle ARIMA(p,d,q) est un modèle telle que $\Delta^d X$ est un processus ARMA(p,q) où Δ est l'opérateur de différentiaition, c'est à dire $(\Delta X)_t = X_t - X_{t-1}$

ARIMA est l'acronyme de AutoRegressive Intergrated Moving Average.

Question 9

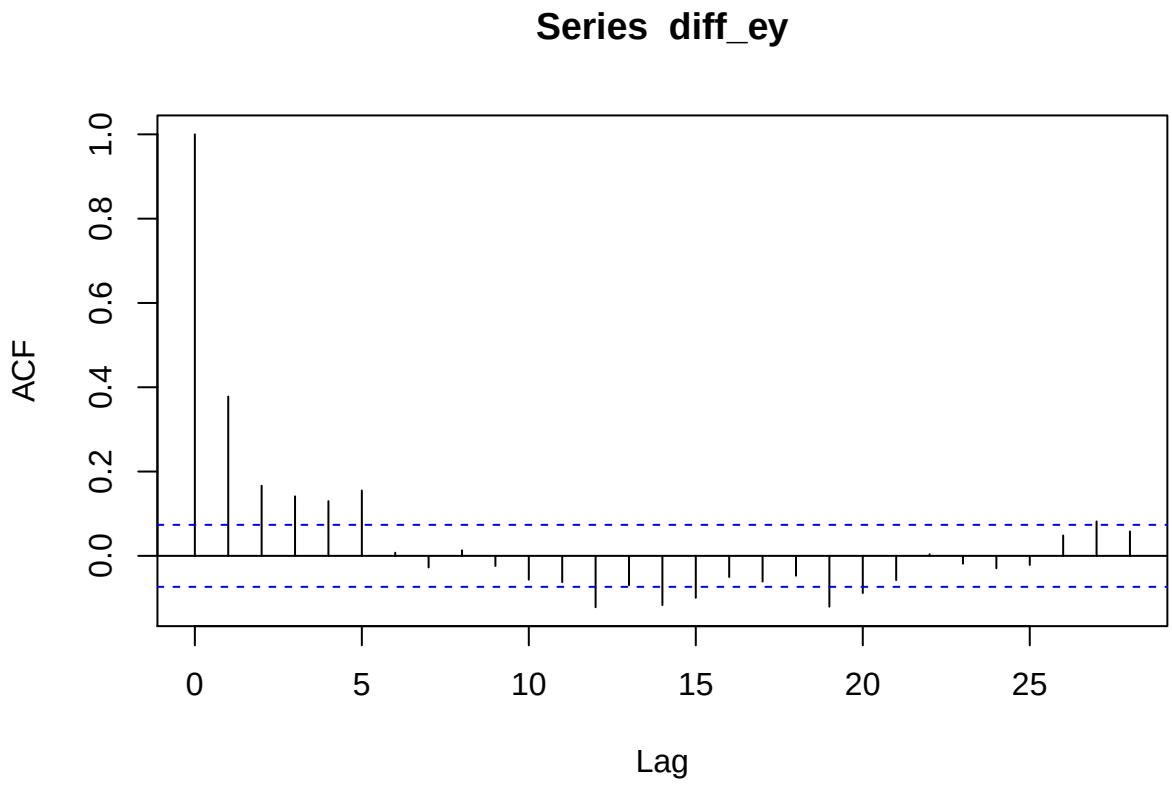
Méthode 1 Pour l'estimation de q, on regarde la fonction d'autocorrélation:

```
acf(x=df$ey)
```



On observe que l'ordre d'autocorrélation est très grand. La série n'est donc pas stationnaire. On considère donc la différenciation au premier ordre :

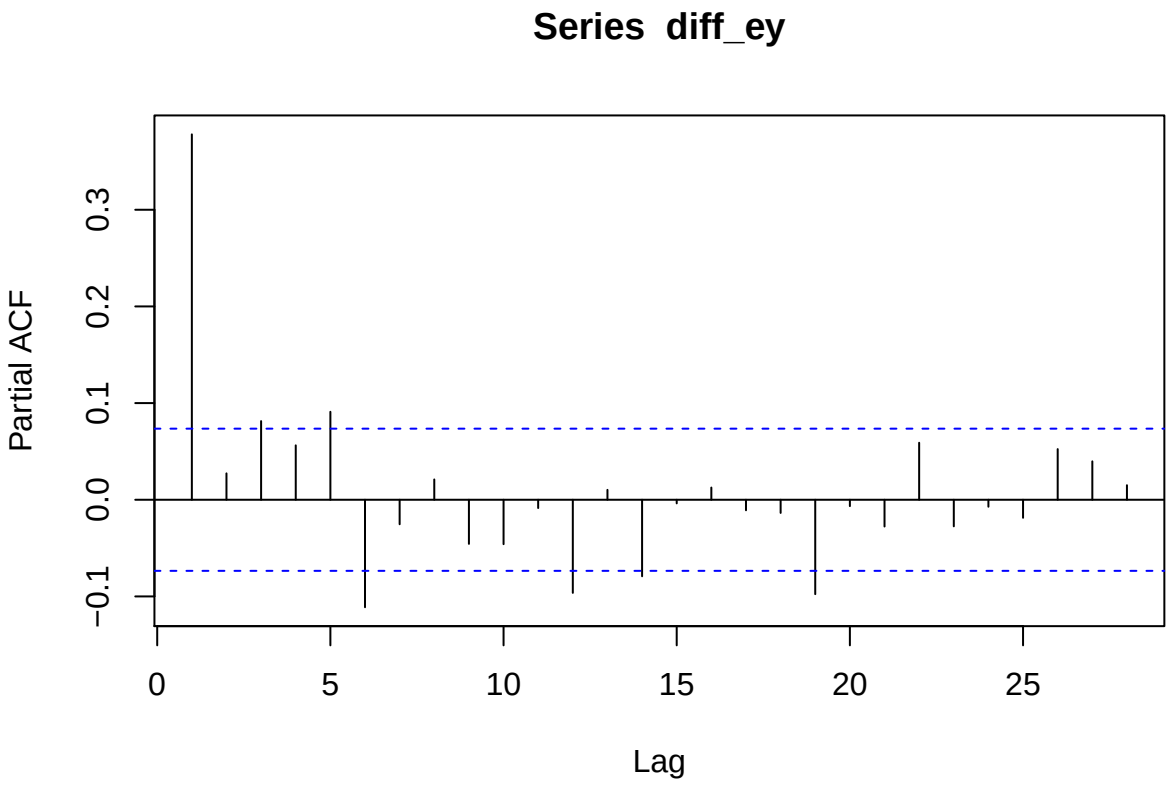
```
diff_ey = diff(df$ey)
acf(diff_ey)
```



On observe des pics jusqu'au 5ème ordre donc on peut estimer que q=5.

Pour estimer p, on regarde la fonction d'autocorrélation partielle :

```
pacf(diff_ey)
```



Il est difficile d'avoir une valeur sûre de p à partir de ce plot, on peut retenir p=1 vu qu'il n'y a pas de pic au deuxième ordre et que les pics suivant ne sont pas importants. Mais on peut supposer à priori que p est égal à 6. On vérifiera cette valeur lors de la méthode 2.

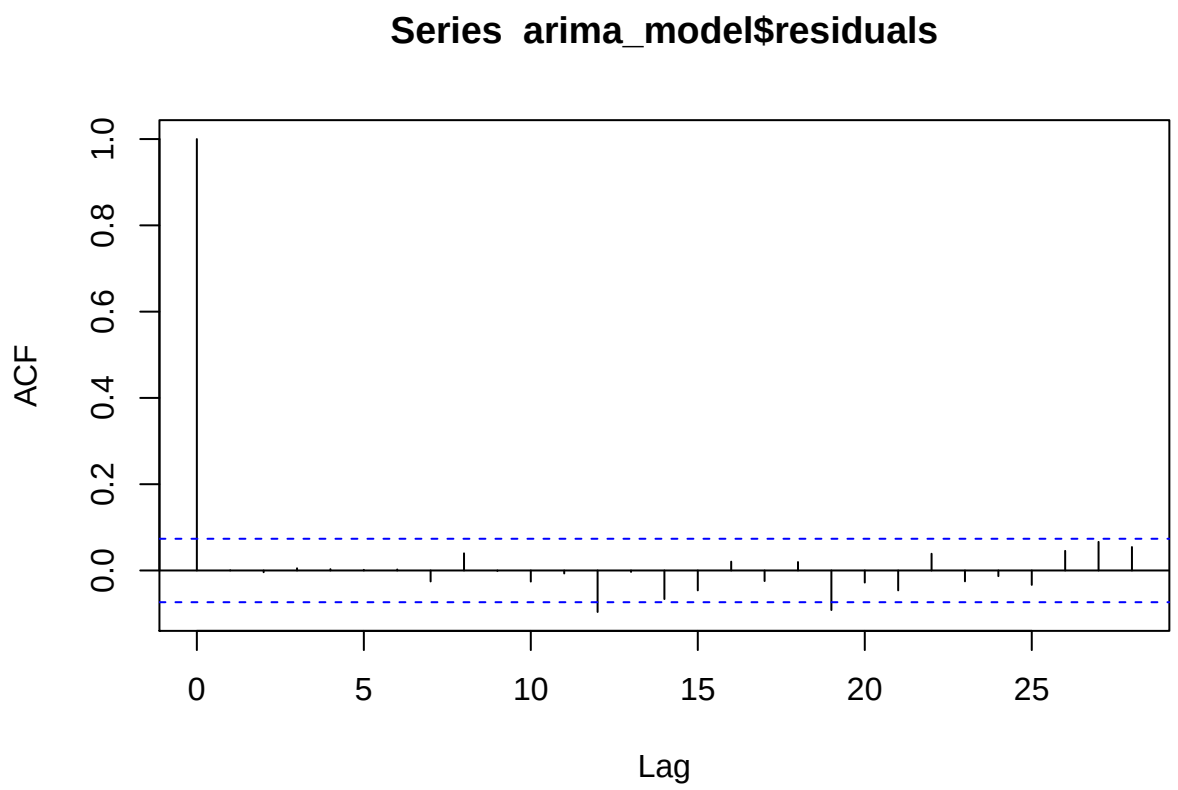
Méthode 2 On utilise la fonction Arima et on regarde les p-values associés aux coefficients du modèle:

```
library(stats)
arima_model = arima(df$ey, order=c(6,1,5))
arima_model
```

```
##
## Call:
## arima(x = df$ey, order = c(6, 1, 5))
##
## Coefficients:
##          ar1      ar2     ar3      ar4     ar5      ar6      ma1      ma2
##       0.8281  -0.0023  0.4753  -0.4278  0.2402  -0.2033  -0.4730  -0.1756
## s.e.  0.1863   0.2585  0.1896   0.2291  0.2239   0.0800   0.1898   0.2271
##           ma3     ma4      ma5
##       -0.4782  0.2603  -0.0365
## s.e.   0.1690  0.2115   0.1901
##
## sigma^2 estimated as 0.06944:  log likelihood = -60.78,  aic = 145.56
```

Le coefficient ar2 n'est pas significatif, et donc on peut retenir un modèle ARIMA(1,1,5) à priori.

```r
arima_model = arima(df$ey, order=c(1,1,5))
acf(arima_model$residuals)
```



L'ACF des résidus correspond bien à une ACF d'un bruit blanc, ce qui montre que notre choix ARIMA(1,1,5) est bien cohérent.

```r
library(tseries)
```

Tests de la stationnarité de la série temporelle

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

```r
tseries::adf.test(df$ey)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  df$ey
## Dickey-Fuller = -3.0444, Lag order = 8, p-value = 0.1362
## alternative hypothesis: stationary
```

Le test d'ADF renvoie une p-value (0.1362) assez grande, ce qui montre qu'il n'y a pas de preuves contre la non-stationnarité de la série temporelle.

```
tseries::kpss.test(df$ey)
```

```
## Warning in tseries::kpss.test(df$ey): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  df$ey
## KPSS Level = 2.9708, Truncation lag parameter = 6, p-value = 0.01
```

Pour le test KPSS, l'hypothèse nulle est que la série est stationnaire. Ici, on a une p-value faible, et donc on peut rejeter l'hypothèse nulle, ce qui confirme que la série est non stationnaire.

```
pp.test(df$ey)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  df$ey
## Dickey-Fuller Z(alpha) = -11.997, Truncation lag parameter = 6, p-value
## = 0.4405
## alternative hypothesis: stationary
```

Le test de Phillips-Perron renvoie une p-value (0.5595) assez grande, ce qui montre qu'il n'y a pas de preuves contre la non-stationnarité de la série temporelle.

Question 10

```
library(forecast)
auto_arima_model = auto.arima(df$ey, max.p = 20,max.q = 20, max.d=3)
auto_arima_model
```

```
## Series: df$ey
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1      ma2
##       0.7451  -0.3803  -0.1295
## s.e.  0.0792   0.0880   0.0496
##
## sigma^2 estimated as 0.07234:  log likelihood=-73.65
## AIC=155.29   AICc=155.35   BIC=173.55
```

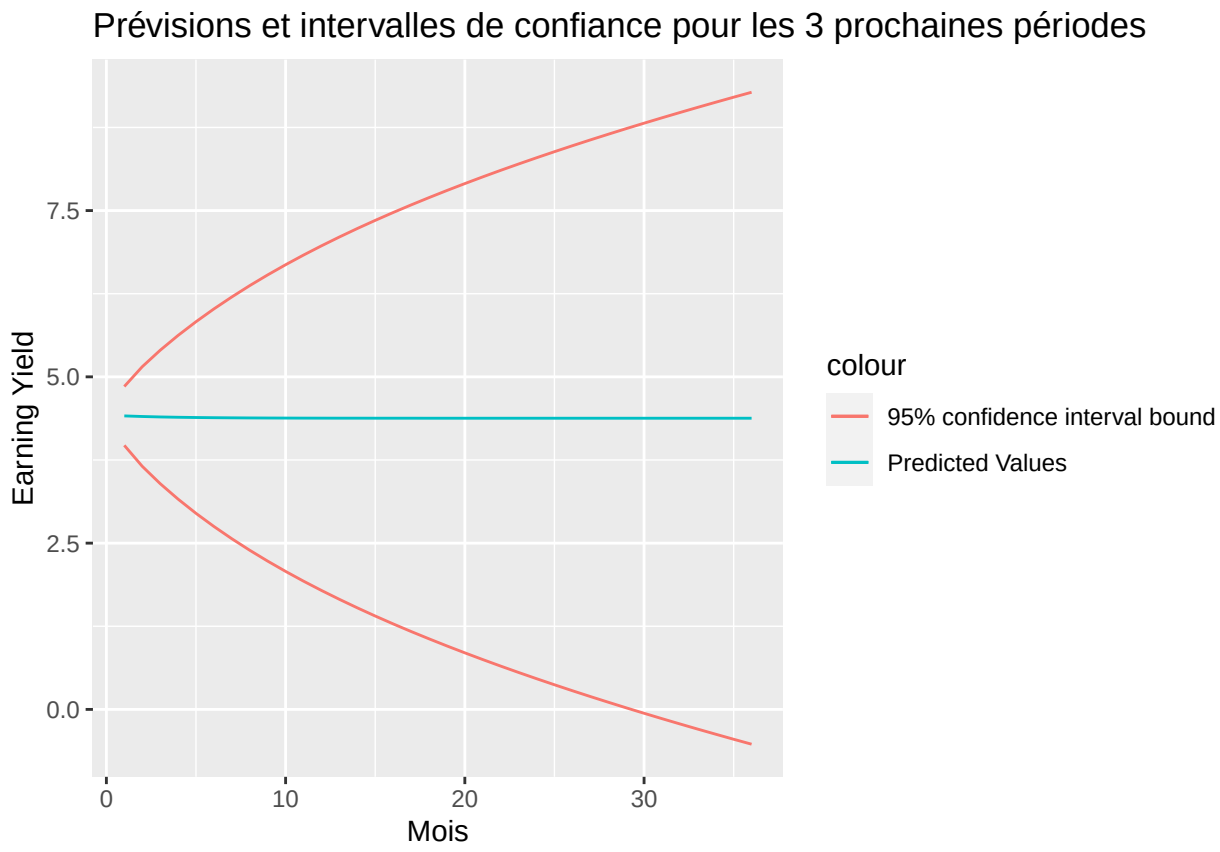
Le modèle finalement retenu est un ARIMA(1,1,2). Notre première estimation ARIMA(1,1,5) a donc été précise pour l'estimation de p et d, mais moins précise pour q.

Question 11

```
predicted_values = stats::predict(auto_arima_model, n.ahead=36)


ggplot()+
  geom_line(aes(x=1:36,y=predicted_values$pred, colour='Predicted Values'))+
  geom_line(aes(x=1:36,y=predicted_values$pred+qnorm(0.95)*predicted_values$se, colour = '95% confidenc
  geom_line(aes(x=1:36,y=predicted_values$pred-qnorm(0.95)*predicted_values$se, colour = '95% confidenc
  labs(title='Prévisions et intervalles de confiance pour les 3 prochaines périodes',
       x='Mois', y='Earning Yield')
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```



Comme prévu pour un modèle ARIMA, les prévisions se stabilisent au cours du temps et convergent vers une valeur fixe. Les intervalles de confiance sont de plus en plus larges.

Question 12

```
library(Metrics)
```

MAE

```
##
## Attaching package: 'Metrics'

## The following object is masked from 'package:forecast':
##
##     accuracy
```

```
scores = data.frame(models=c('Manual ACF/PACF Model', 'AutoArima Model'),
                    MAE = c(mae(df$ey, df$ey - arima_model$residuals),
                            mae(df$ey, df$ey - auto_arima_model$residuals)),
                    RMSE = c(rmse(df$ey, df$ey - arima_model$residuals),
                             rmse(df$ey, df$ey - auto_arima_model$residuals)))
print(scores)
```

```
##                  models       MAE      RMSE
## 1 Manual ACF/PACF Model 0.1764164 0.2652843
## 2       AutoArima Model 0.1757934 0.2681952
```

Les MAE et RMSE sont équivalents pour les deux modèles. Le premier est meilleur en terme de MAE et le deuxième est un peu meilleur un terme de RMSE.

Question 11.bis

```
resid1 = arima_model$residuals
resid2 = auto_arima_model$residuals
d = resid1^2 - resid2^2
d.cov <- acf(d, na.action = na.omit, lag.max = 0, type = "covariance", plot = FALSE)$acf[, , 1]
d.var <- sum(c(d.cov[1], 2 * d.cov[-1]))/length(d)
dv <- d.var
stat <- mean(d, na.rm = TRUE)/sqrt(dv)
print(stat)
```

```
## [1] -1.320548
```

$|S| < 1.96$ et donc on retient l'hypothèse nulle, d'où les deux modèles ont presque le même pouvoir prédictif.

Partie 4: Stabilité du modèle

Question 13

```
n_obs = nrow(df)
betas = c()
lower_bound = c()
upper_bound = c()

for (i in 1:(n_obs-60)){
  sub_df = df[i:(i+59),]
  sub_model = lm(ey~rates, data=sub_df)
```

```
  coefs = summary(sub_model)$coefficients
  betas = append(betas, coefs[2,1])
  lower_bound = append(lower_bound, coefs[2,1]-qnorm(0.95)*coefs[2,2])
  upper_bound = append(upper_bound, coefs[2,1]+qnorm(0.95)*coefs[2,2])}

ggplot() +
  geom_line(aes(x=df[1:(n_obs-60), 'date'], y=betas, colour='beta'))+
  geom_line(aes(x=df[1:(n_obs-60), 'date'], y=lower_bound, colour='95% Confidence interval'))+
  geom_line(aes(x=df[1:(n_obs-60), 'date'], y=upper_bound, colour='95% Confidence interval'))+
  labs(title = 'Evolution of beta', x='First observation year', y='beta')
```

On remarque une instabilité claire de l'estimation de beta. Cette instabilité est plus importante à partir de la fin des années 90, où beta alterne carrément entre valeurs positives et négatives, et donc on a à la fois des périodes où le Earning Yield croit avec taux d'intérêt sans risque et d'autres périodes où il décroit quand les taux d'intérêt croissent.

Question 14

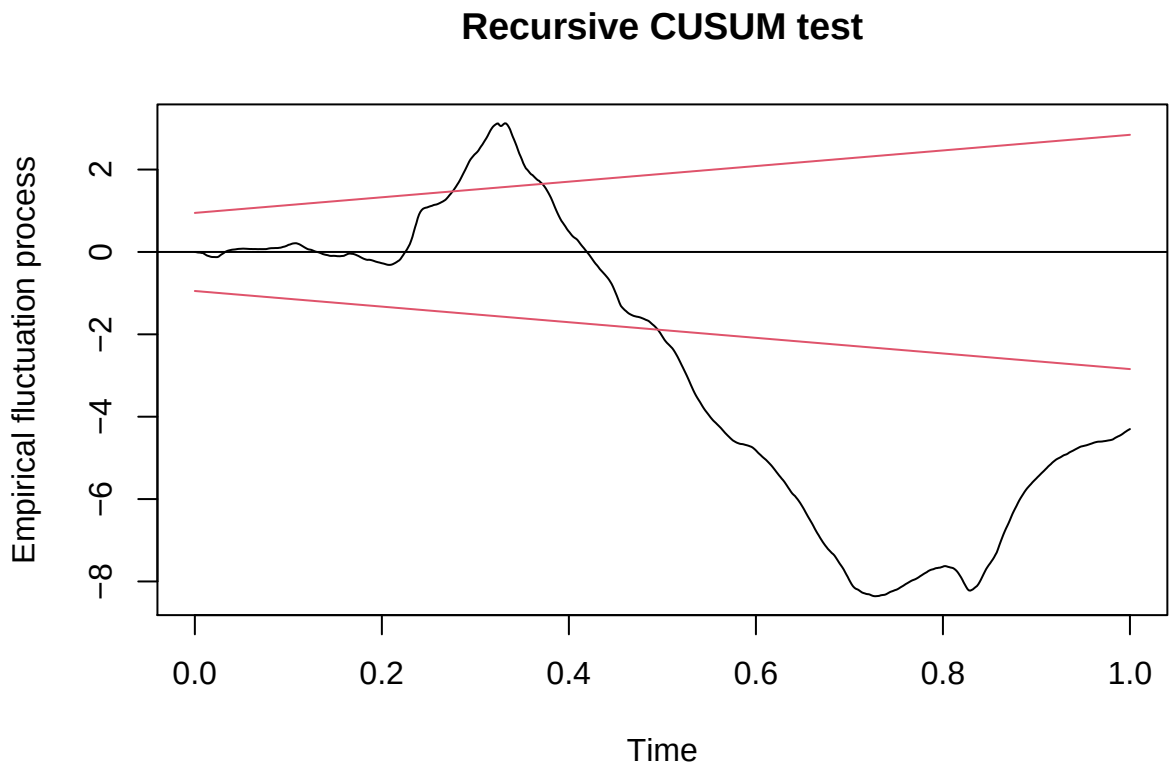
Le test de CUSUM Le test de CUSUM permet de détecter les instabilités des équations de régression au cours du temps.Pour cela, il se base sur la somme cumulée des résidus récursifs, d'où son nom (CUmulative SUM). Pour ce test, on trace l'évolution de la courbe de la somme cumulée au cours du temps et si cette courbe ne coupe pas une certaine frontière (appelée corridor) alors le modèle est stable. L'hypothèse nulle de ce test est que le modèle est stable.

Implémentation D'abord, on trace les résidus cumulatifs et les frontières (en rouge) du CUSUM test.

```
library(strucchange)
```

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

```
cusum_plot <- efp(ey~rates, type="Rec-CUSUM", data=df)
plot(cusum_plot)
```



On remarque que la courbe CUSUM coupe et s'éloigne largement du corridor ce qui suggère que le modèle est instable. On regarde ensuite la p-value du test CUSUM pour confirmer:

```
sctest(ey~rates, data = df)
```

```
##
##  Recursive CUSUM test
##
## data:  ey ~ rates
## S = 3.4084, p-value < 2.2e-16
```

La p-value du test est très faible et donc on peut rejeter l'hypothèse null (H0: le modèle est stable), ce qui confirme donc l'instabilité du modèle.