Econometry new

2025-01-25

Importation des données

```
#importation des différentes librairies nécessaires pour la suite du projet
library(ggplot2)
library(cowplot)
library(car)
```

```
## Loading required package: carData
```

```
library(carData)
library(caret)
```

```
## Loading required package: lattice
```

```
library(FactoMineR)
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:car':
##
##     recode
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(lmtest) #pour Breusch-Pagan
```

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

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

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

```
library(skedastic) #pour White
library(nortest) #pour Anderson-Darling
library(olsrr) #pour White
```

```
## Warning: package 'olsrr' was built under R version 4.3.3
```

```
##
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
#ouverture des jeux de données de consommation d'électricité des ménages
celec<-read_excel("celec_menages.xlsx", col_names = TRUE)
```

```
#nous regardons la structure des données de la table "celec"
str(celec)
```

```
## tibble [32 x 7] (S3: tbl_df/tbl/data.frame)
##  $ Date         : num [1:32] 1990 1991 1992 1993 1994 ...
##  $ IPC          : num [1:32] 67.4 69.6 71.2 72.7 73.9 75.3 76.8 77.7 78.2 78.6 ...
##  $ PIB2020      : num [1:32] 1566 1586 1610 1604 1642 ...
##  $ Pelec        : num [1:32] 125 121 124 126 127 ...
##  $ Pop1         : num [1:32] 56708831 56975597 57239847 57467085 57658772 ...
##  $ DJU          : num [1:32] 0.96 1.167 1.076 1.076 0.922 ...
##  $ Celec_menages: num [1:32] 96.9 106.8 109.6 111.5 111.2 ...
```

IPC : indice annuel des prix à la consommation PIB2020 : produit intérieur brut en euros de 2020 Pelec : Prix de l'électricité des ménages (euro/MWh) Pop1 : population France métropolitaine DJU : Indice de rigueur du climat (MTES) Celec_menages : consommation électrice GWh observée

#%==========%= ## PREPARING SET FOR EXPLORATION AND REGRESSION #### #%==========

Renommer les colonnes pour une meilleure clarté

IRC = Indice de Rigueur Climatique.

```
celec <- celec %>%
 rename(
       IRC = DJU,
       Population = Pop1,
       elec_cons = Celec_menages,
   )
```

Synthèse descriptive des données

```
summary(celec)
```

```
##       Date           IPC            PIB2020         Pelec
##  Min.   :1990   Min.   : 67.40   Min.   :1566   Min.   :109.9
##  1st Qu.:1998   1st Qu.: 78.08   1st Qu.:1792   1st Qu.:114.1
##  Median :2006   Median : 88.61   Median :2149   Median :124.5
##  Mean   :2006   Mean   : 88.12   Mean   :2065   Mean   :133.6
##  3rd Qu.:2013   3rd Qu.: 99.58   3rd Qu.:2303   3rd Qu.:148.4
##  Max.   :2021   Max.   :106.45   Max.   :2505   Max.   :193.1
##    Population            IRC           elec_cons
##  Min.   :56708831   Min.   :0.8311   Min.   : 96.91
##  1st Qu.:58350214   1st Qu.:0.9322   1st Qu.:122.49
##  Median :61389492   Median :1.0034   Median :142.46
##  Mean   :61214776   Mean   :0.9995   Mean   :138.43
```

```
##  3rd Qu.:63938216   3rd Qu.:1.0475   3rd Qu.:156.87
##  Max.   :65613522   Max.   :1.1943   Max.   :166.67
```

#%========== ## Going base 100 for all variables in 2015

```
base_2015 <- celec %>% filter(Date == 2015)
```

```
celec <- celec %>%
    mutate(
        PIB2020_base100_2015 = PIB2020 / base_2015$PIB2020 * 100,
        Pelec_base100_2015 = Pelec / base_2015$Pelec * 100,
        Population_base100_2015 = Population / base_2015$Population * 100,
        IRC_base100_2015 = IRC / base_2015$IRC * 100,
        elec_cons_base100_2015 = elec_cons / base_2015$elec_cons * 100,
        IPC_base100_2015 = IPC  # L'IPC est déjà en base 100
    )
```

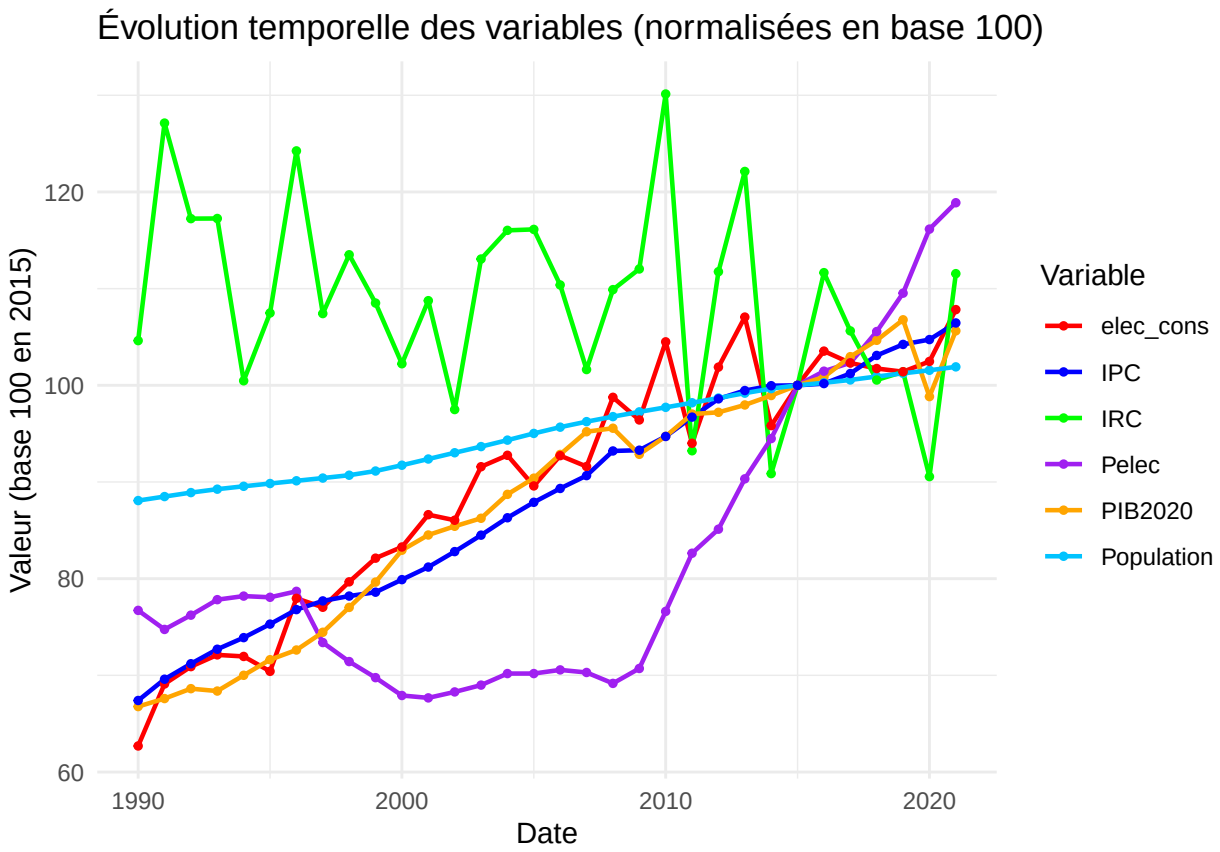
Tracer toutes les colonnes qui se terminent par base100_2015 en fonction de Date

Convertir les données en format long pour ggplot

```
celec_long <- celec %>%
    select(Date, ends_with("base100_2015")) %>%
    pivot_longer(cols = -Date, names_to = "variable", values_to = "value") %>%
    mutate(variable = gsub("_base100_2015", "", variable))
```

Tracer les données

```
ggplot(celec_long, aes(x = Date, y = value, color = variable)) +
    geom_line(linewidth = 0.8) +
    geom_point(size = 1) +
    scale_color_manual(values = c("red", "blue", "green", "purple", "orange", "#00c3ff")) +
    labs(title = "Évolution temporelle des variables (normalisées en base 100)",
       x = "Date",
       y = "Valeur (base 100 en 2015)",
       color = "Variable") +
    theme_minimal()
```



#%========== # Adding new variables for the regression

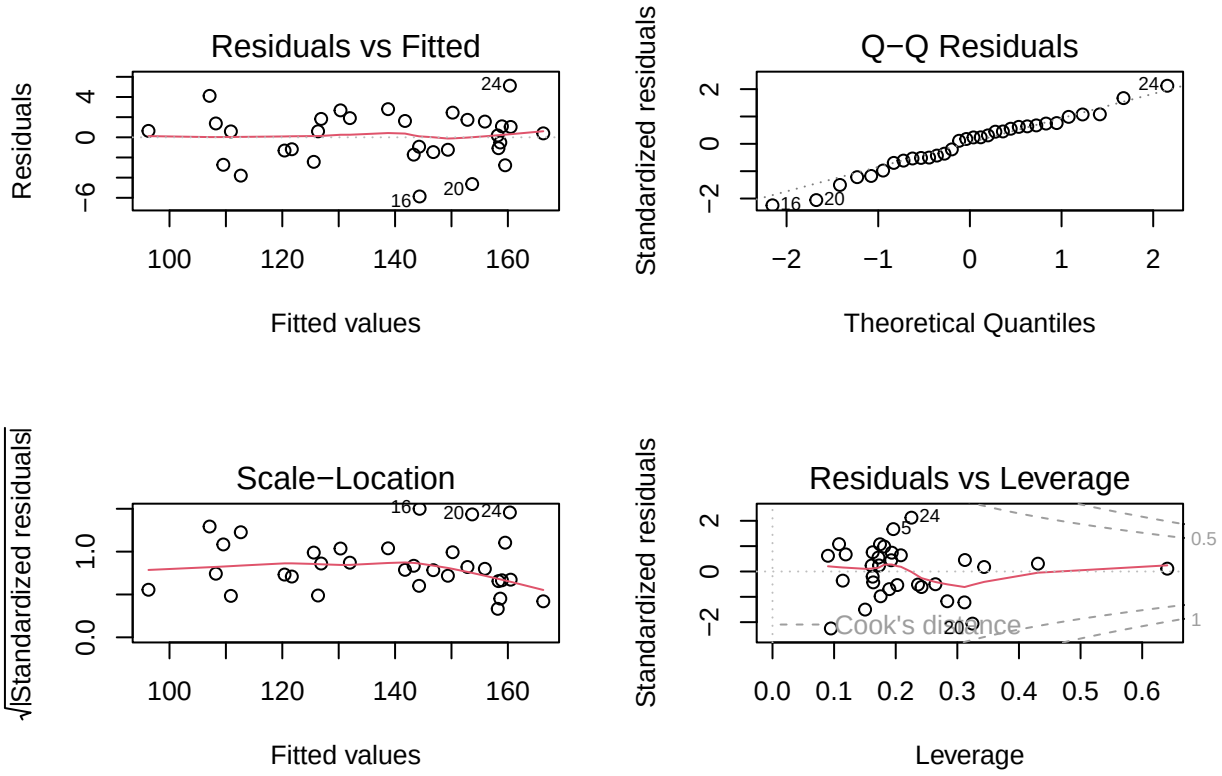
Le PIB est en euro constant 2020 mais l'inflation en base 2015 : on doit ajuster le PIB en 2015

```
celec <- celec %>%
    mutate(
        elec_cons_pc = elec_cons / Population, # pc = per capita
        PIB2015 = PIB2020 * (IPC[Date == 2015] / IPC[Date == 2020]), #en milliards d'euros 2015
        PIB2015_pc = PIB2015 / Population, #in 2015 10^9 euros per capita,
        Pelec_euro2015 = Pelec * (IPC[Date == 2015] / IPC), # Prix de l'électricité en euro constant 20
    )
```

#%==========
PREMIERE REGRESSION GLOBALE SANS PRISE EN COMPTE DE LA RUPTURE EN 2009
#%==========

Régression

```
celec.lm=lm(elec_cons~PIB2015 + Population + IRC + Pelec_euro2015 + IPC + Date, data = celec)
par(mfrow=c(2,2))
plot(celec.lm)
```



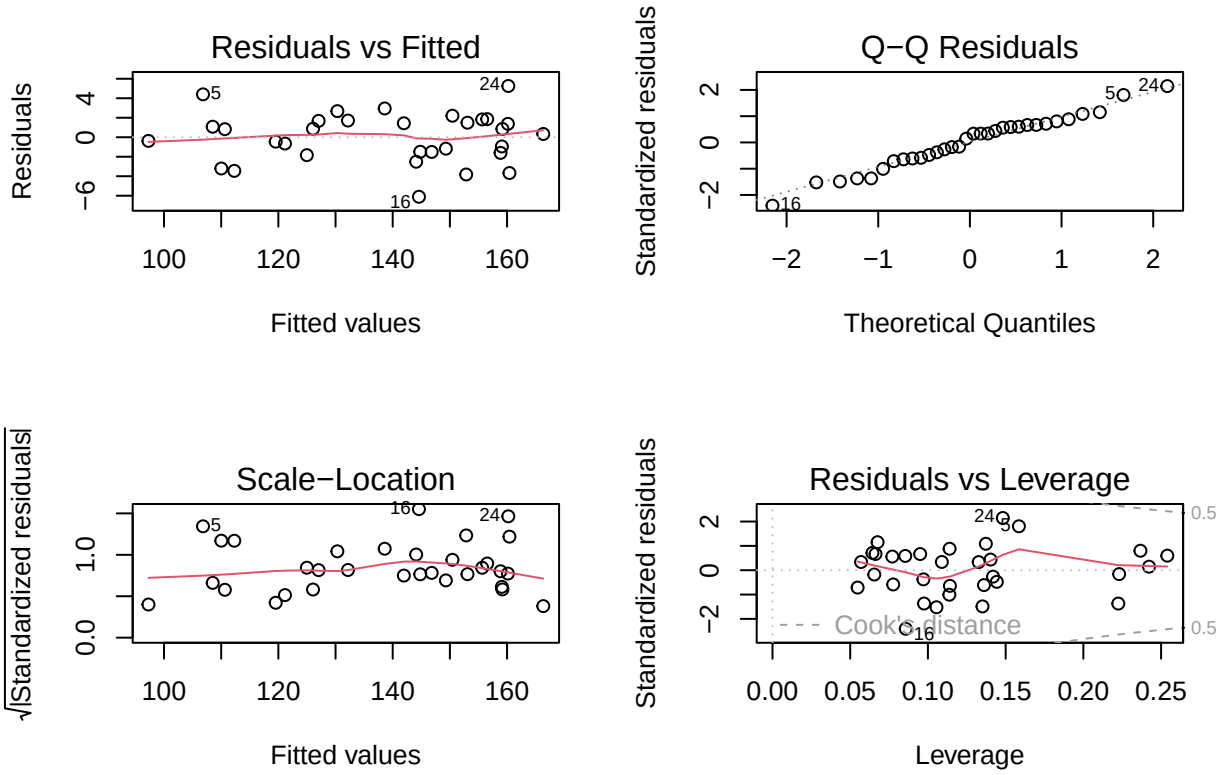

```
summary(celec.lm)
```

```
##
## Call:
## lm(formula = elec_cons ~ PIB2015 + Population + IRC + Pelec_euro2015 +
##     IPC + Date, data = celec)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8659 -1.3512  0.4899  1.6491  5.1223
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -4.905e+03  1.531e+03  -3.205  0.00367 **
## PIB2015        -1.389e-02  1.479e-02  -0.939  0.35671
## Population     -3.636e-07  2.591e-06  -0.140  0.88953
## IRC             4.093e+01  6.044e+00   6.771 4.26e-07 ***
## Pelec_euro2015 -2.348e-01  4.814e-02  -4.876 5.13e-05 ***
## IPC             8.673e-02  7.253e-01   0.120  0.90577
## Date            2.533e+00  7.893e-01   3.210  0.00363 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.742 on 25 degrees of freedom
## Multiple R-squared:  0.9847, Adjusted R-squared:  0.981
## F-statistic: 267.5 on 6 and 25 DF,  p-value: < 2.2e-16
```

Les variables PIB2015, Population et IPC ont des p-value supérieures à 5% donc nous les retirons.

Deuxième régression

```
celec.lm2=lm(elec_cons~IRC + Pelec_euro2015 + Date, data = celec)
par(mfrow=c(2,2))
plot(celec.lm2)
```



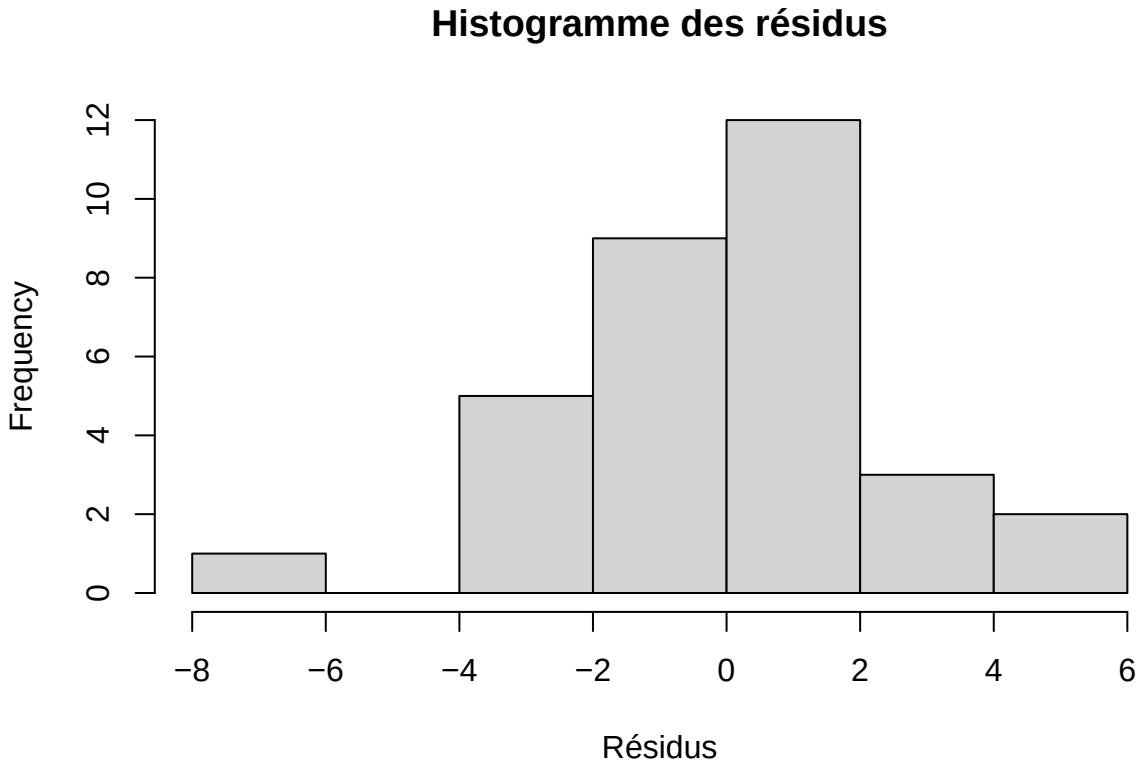

```
summary(celec.lm2)
```

```
##
## Call:
## lm(formula = elec_cons ~ IRC + Pelec_euro2015 + Date, data = celec)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1082 -1.5322  0.5801  1.6865  5.2570
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -4.124e+03  1.116e+02 -36.967  < 2e-16 ***
## IRC             4.122e+01  5.700e+00   7.231 7.16e-08 ***
## Pelec_euro2015 -2.021e-01  2.447e-02  -8.258 5.49e-09 ***
## Date            2.120e+00  5.434e-02  39.019  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.654 on 28 degrees of freedom
## Multiple R-squared:  0.9839, Adjusted R-squared:  0.9822
## F-statistic: 570.5 on 3 and 28 DF,  p-value: < 2.2e-16
```

Vérification de la normalité des résidus

Histogramme des résidus

```
hist(residuals(celec.lm2), main="Histogramme des résidus", xlab="Résidus")
```



Test de Shapiro-Wilk

```
shapiro.test(residuals(celec.lm2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(celec.lm2)
## W = 0.9807, p-value = 0.8196
```

Normalité validée

Test d'Anderson-Darling

```
ad.test(residuals(celec.lm2))
```

```
##
##  Anderson-Darling normality test
##
## data:  residuals(celec.lm2)
## A = 0.31579, p-value = 0.5251
```

Normalité validée

Vérification de l'hétéroscédasticité

Test de Breusch-Pagan

```
bptest(celec.lm2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  celec.lm2
## BP = 1.8721, df = 3, p-value = 0.5994
```

Homoscédasticité validée

Test de White

```
bptest(celec.lm2, ~ fitted(celec.lm2) + I(fitted(celec.lm2)))
```

```
##
##  studentized Breusch-Pagan test
##
## data:  celec.lm2
## BP = 0.12413, df = 1, p-value = 0.7246
```

Test d'autocorrélation

##Test de Breusch-Godfrey

```
bgtest(celec.lm2)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  celec.lm2
## LM test = 0.038089, df = 1, p-value = 0.8453
```

Pas d'autocorrélation.

Etude de la multicolinéarité

```
vif(celec.lm2)
```

```
##            IRC Pelec_euro2015           Date
##       1.147436       1.020445       1.143243
```

VIF pas élevé (inférieur à 5) donc pas d'autocorrelation notable.

```
vif(celec.lm)
```

```
##        PIB2015     Population            IRC Pelec_euro2015            IPC
##      73.898719     239.556870       1.209056       3.701178     304.932156
##           Date
##     226.017040
```

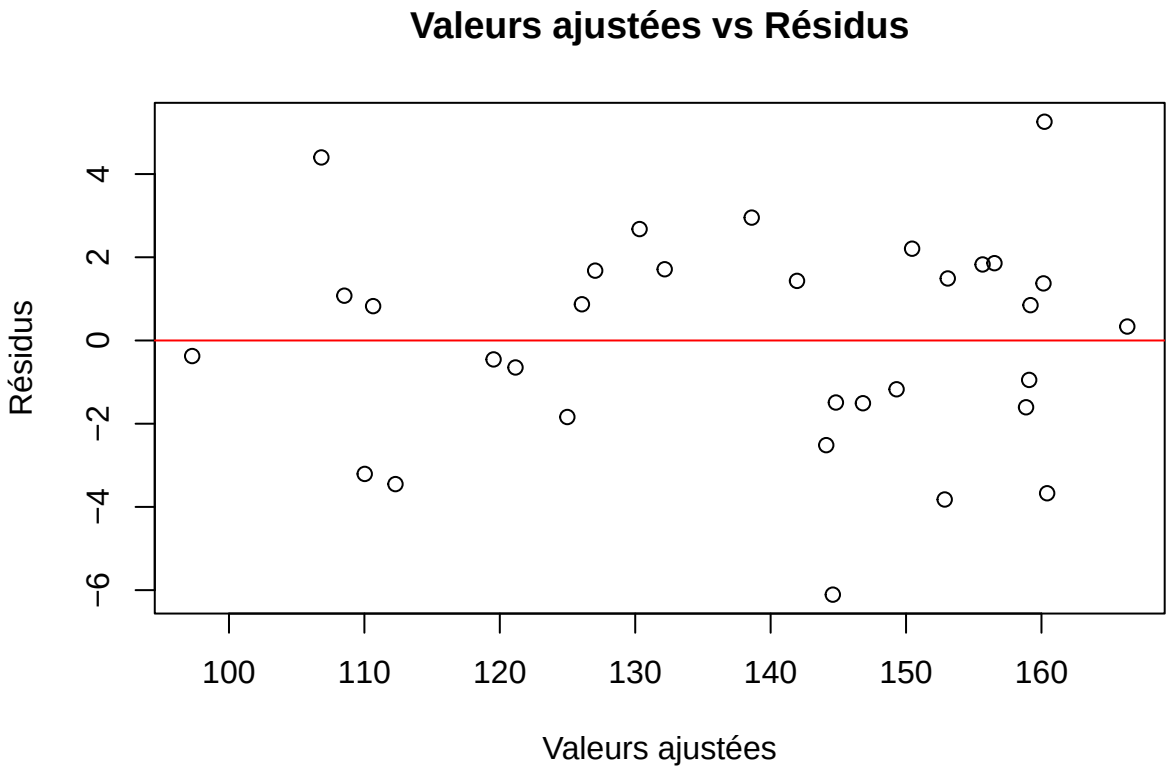
Dans le premier modèle, les VIF étaient très élevées.

-> Donc pas besoin de passer par du Lasso et de l'ACP.

Diagnostic visuel

Graphique des valeurs ajustées vs résidus

```
plot(fitted(celec.lm2), residuals(celec.lm2), main="Valeurs ajustées vs Résidus", xlab="Valeurs ajustées
abline(h=0, col="red")
```



Perfetto

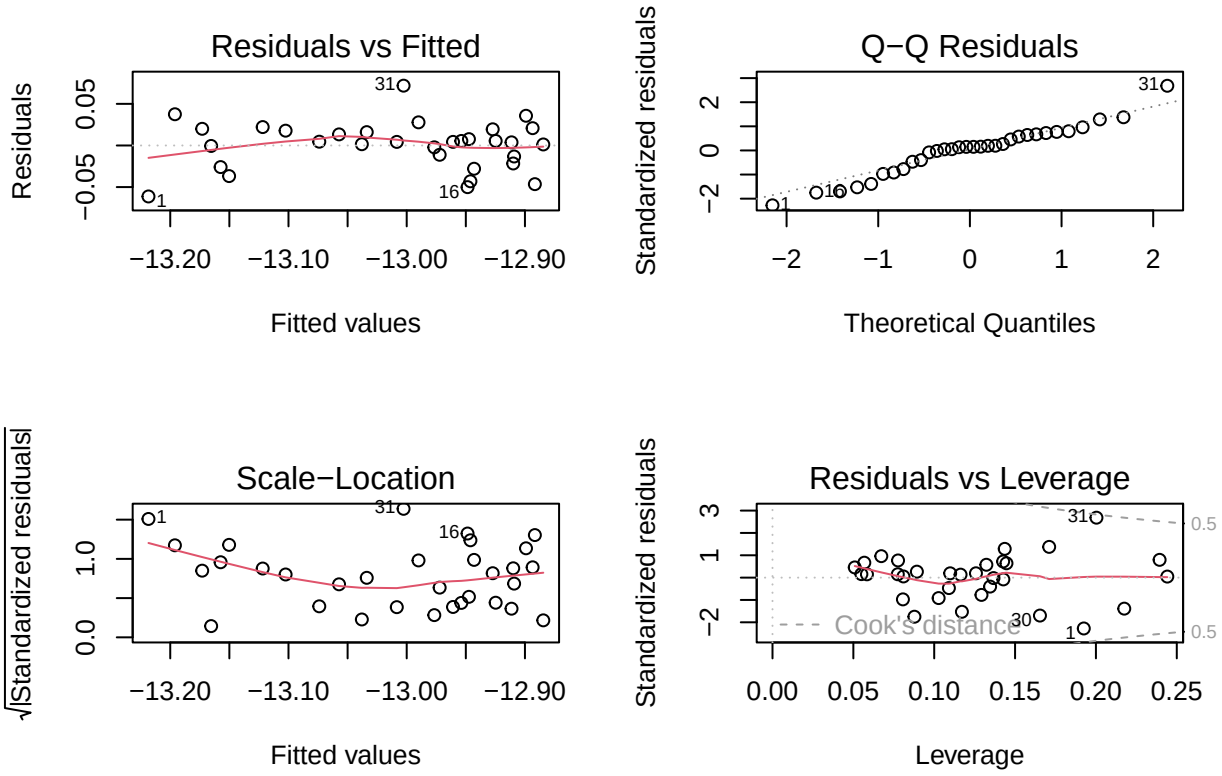
QQ-plot des résidus

```
qqnorm(residuals(celec.lm2))
qqline(residuals(celec.lm2), col="red")
```

ça ser-
pente un peu quand même.

#%===%=
DEUXIEME MODELE AVEC PRISE EN COMPTE DE LA RUPTURE EN 2009
#%===%=

```
celec.lm3=lm(log(elec_cons_pc)~log(PIB2015_pc) + log(Pelec_euro2015) + IRC , data = celec)
par(mfrow=c(2,2))
plot(celec.lm3)
```




```
summary(celec.lm3)
```

```
##
## Call:
## lm(formula = log(elec_cons_pc) ~ log(PIB2015_pc) + log(Pelec_euro2015) +
##     IRC, data = celec)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.061256 -0.015346  0.004301  0.018180  0.071845
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -2.70320    0.54414  -4.968 3.03e-05 ***
## log(PIB2015_pc)       1.00773    0.05931  16.992 2.78e-16 ***
## log(Pelec_euro2015) -0.03134    0.04537  -0.691 0.495441
## IRC                   0.28438    0.06487   4.384 0.000149 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02995 on 28 degrees of freedom
## Multiple R-squared:  0.9279, Adjusted R-squared:  0.9202
## F-statistic: 120.2 on 3 and 28 DF,  p-value: 4.273e-16
```

test de Chow pas immédiatement applicable ? -> les résidus du modèle doivent être indépendants et ne pas montrer de tendance -> or les graphs sont un peu dégueux ?

```
# Création de deux groupes pour application du test de Chow
groupe1 <- subset(celec, Date<2009)
groupe2 <- subset(celec, Date>=2009)
```

```r
lm1_chow <- lm(log(elec_cons_pc) ~ log(PIB2015_pc) + log(Pelec_euro2015) + IRC, data = groupe1)
lm2_chow <- lm(log(elec_cons_pc) ~ log(PIB2015_pc) + log(Pelec_euro2015) + IRC, data = groupe2)
```

```r
#Ajout de l'indication groupe 1 et groupe 2 dans la table celec
celec$group <- ifelse(celec$Date < 2009, "groupe1", "groupe2")
```

```r
lm_global <- lm(log(elec_cons_pc) ~ log(PIB2015_pc) * group +
                                    log(Pelec_euro2015) * group +
                                    IRC * group,
                data = celec)
```

```r
# Test de Chow
linearHypothesis(lm_global,
                 c("log(PIB2015_pc):groupgroupe2 = 0",
                   "groupgroupe2:log(Pelec_euro2015) = 0",
                   "groupgroupe2:IRC = 0"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## log(PIB2015_pc):groupgroupe2 = 0
## groupgroupe2:log(Pelec_euro2015) = 0
## groupgroupe2:IRC = 0
##
## Model 1: restricted model
## Model 2: log(elec_cons_pc) ~ log(PIB2015_pc) * group + log(Pelec_euro2015) *
##     group + IRC * group
##
##   Res.Df       RSS Df Sum of Sq      F    Pr(>F)
## 1     27 0.0229376
## 2     24 0.0093388  3  0.013599 11.649 6.603e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

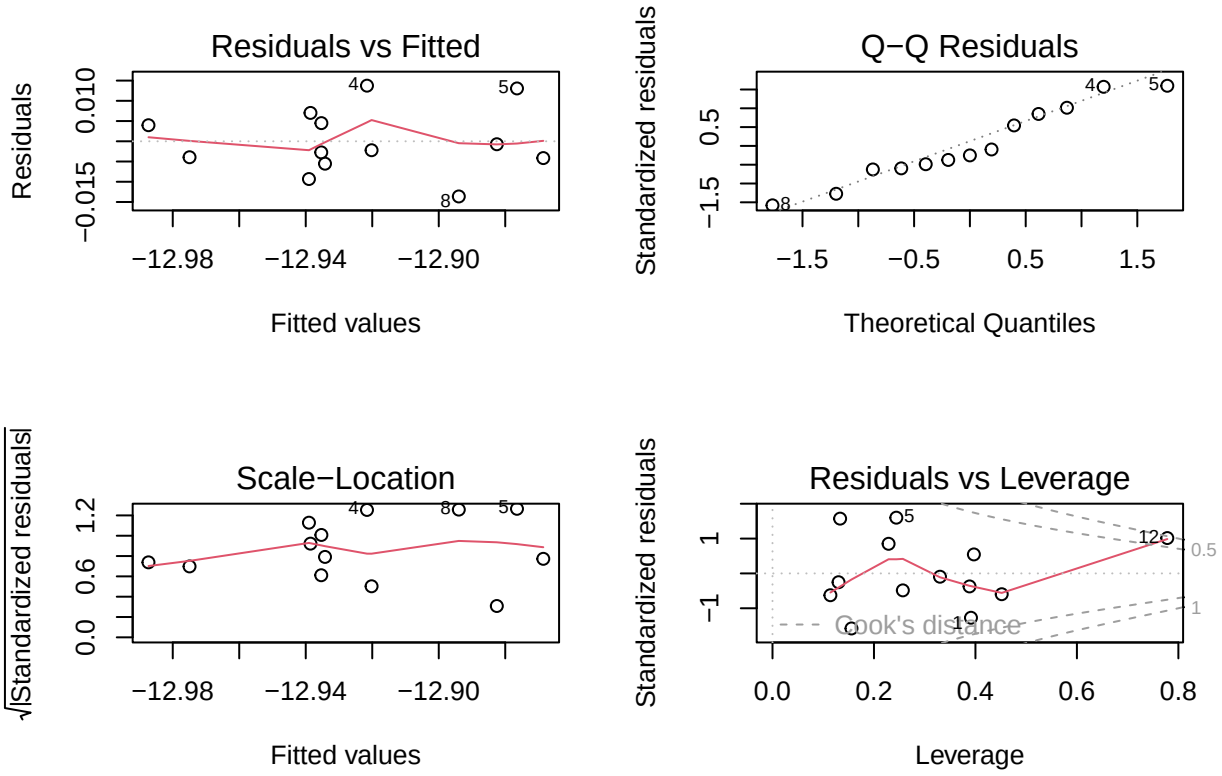
p-value inférieure à 0.05, donc valide la rupture en 2009

On crée la table qui ne contient que les données supérieures à 2009

```r
celec_2009 <- celec[celec$Date >= 2009, ]
```

Modele de régression à partir de 2009

```r
celec.lm4=lm(log(elec_cons_pc)~log(PIB2015_pc) + log(Pelec_euro2015) + IRC, data = celec_2009)
par(mfrow=c(2,2))
plot(celec.lm4)
```




```
summary(celec.lm4)
```

```
##
## Call:
## lm(formula = log(elec_cons_pc) ~ log(PIB2015_pc) + log(Pelec_euro2015) +
##     IRC, data = celec_2009)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.013633 -0.004151 -0.002211  0.004489  0.013742
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -17.55915    1.48111 -11.855 8.54e-07 ***
## log(PIB2015_pc)      -0.31061    0.13127  -2.366 0.042169 *
## log(Pelec_euro2015)   0.21582    0.03476   6.208 0.000157 ***
## IRC                   0.36781    0.02792  13.172 3.47e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.00939 on 9 degrees of freedom
## Multiple R-squared:  0.9509, Adjusted R-squared:  0.9346
## F-statistic: 58.12 on 3 and 9 DF,  p-value: 3.26e-06
```

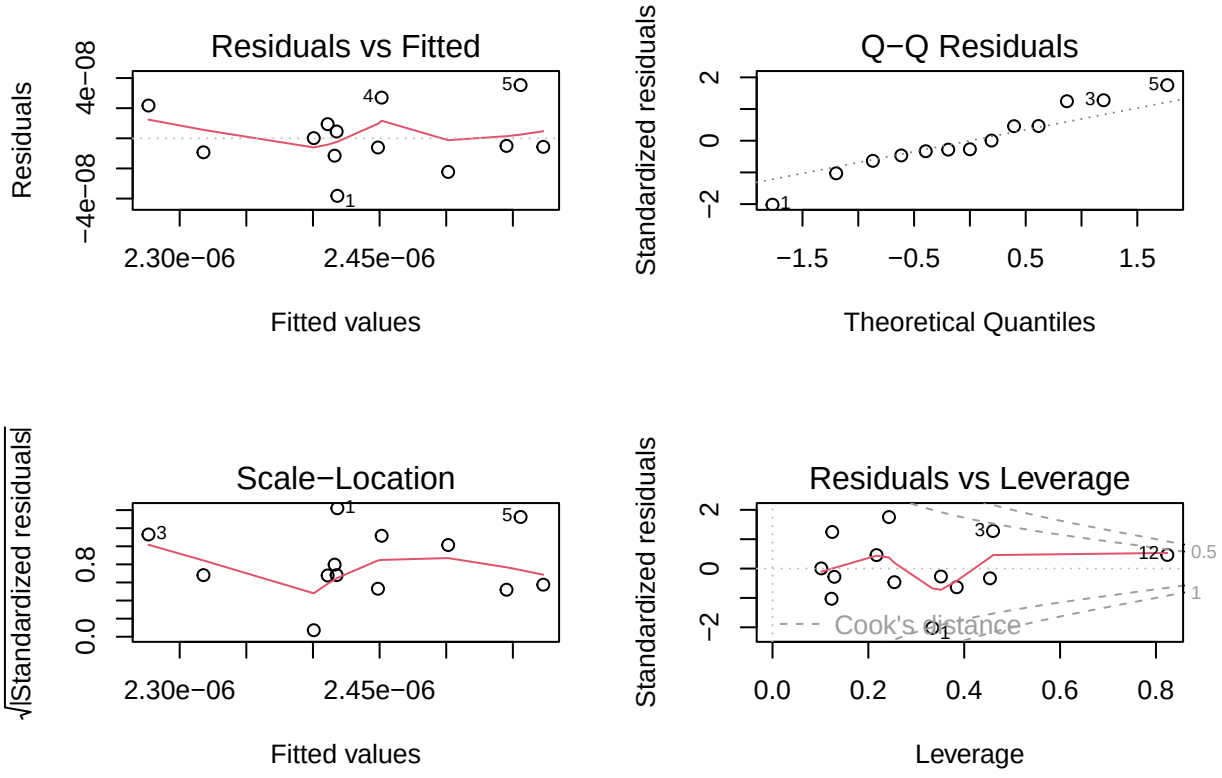
```
step(celec.lm4)
```

```
## Start:  AIC=-118.15
## log(elec_cons_pc) ~ log(PIB2015_pc) + log(Pelec_euro2015) + IRC
##
##                       Df Sum of Sq       RSS      AIC
```

```
## <none>                              0.0007936 -118.150
## - log(PIB2015_pc)      1 0.0004937 0.0012873 -113.862
## - log(Pelec_euro2015)  1 0.0033985 0.0041921  -98.513
## - IRC                  1 0.0152999 0.0160935  -81.026
```

```
##
## Call:
## lm(formula = log(elec_cons_pc) ~ log(PIB2015_pc) + log(Pelec_euro2015) +
##     IRC, data = celec_2009)
##
## Coefficients:
##         (Intercept)      log(PIB2015_pc)  log(Pelec_euro2015)
##             -17.5592              -0.3106               0.2158
##                 IRC
##              0.3678
```

```
celec.lm5=lm(elec_cons_pc~ IRC + PIB2015_pc + Population, data = celec_2009)
par(mfrow=c(2,2))
plot(celec.lm5)
```




```
summary(celec.lm5)
```

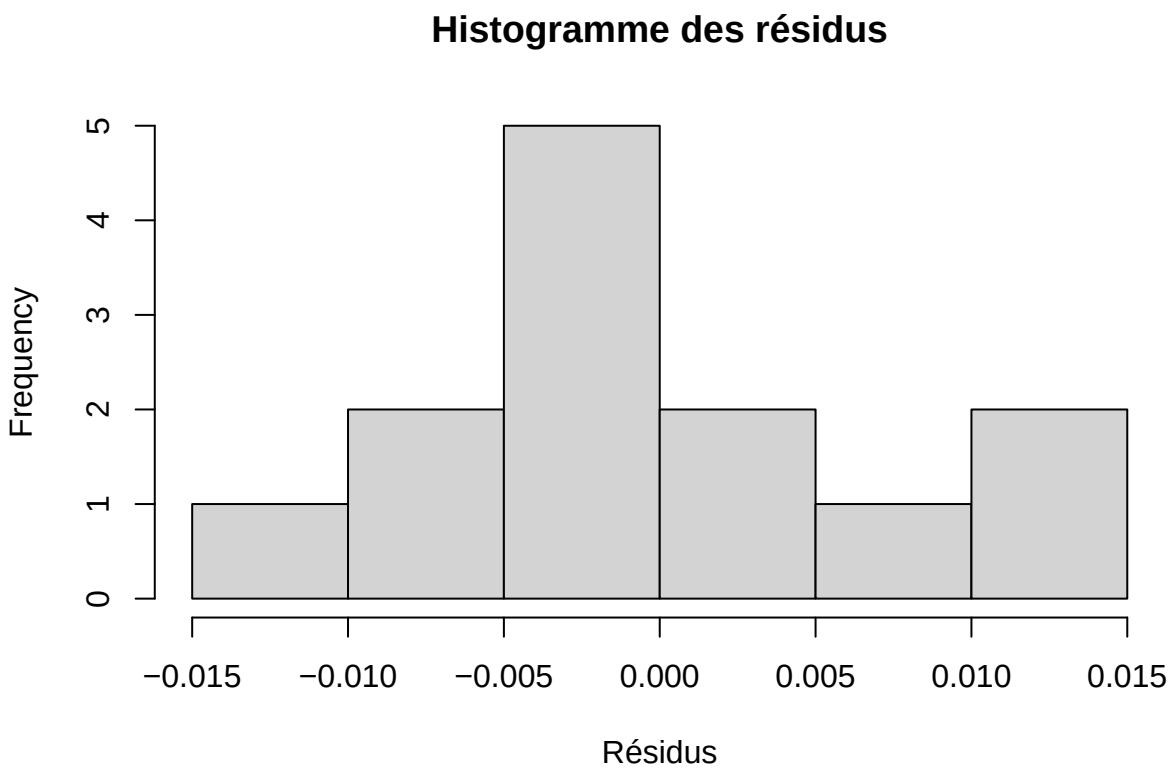
```
##
## Call:
## lm(formula = elec_cons_pc ~ IRC + PIB2015_pc + Population, data = celec_2009)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.819e-08 -9.283e-09 -5.038e-09  9.407e-09  3.534e-08
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.922e-06  5.582e-07  -3.443 0.007355 **
## IRC          8.854e-07  6.803e-08  13.015 3.85e-07 ***
## PIB2015_pc  -2.662e-02  9.784e-03  -2.720 0.023590 *
## Population   6.890e-14  1.128e-14   6.109 0.000177 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.317e-08 on 9 degrees of freedom
## Multiple R-squared:  0.9499, Adjusted R-squared:  0.9332
## F-statistic: 56.92 on 3 and 9 DF,  p-value: 3.561e-06
```

Vérification de la normalité des résidus

Histogramme des résidus

```
hist(residuals(celec.lm4), main="Histogramme des résidus", xlab="Résidus")
```



Visually, there's a hint of non-normality, as a further increase can be seen on the right-hand side of the graph. However, the data still has a Gaussian shape, so we need to examine the normality of the residuals further. The Shapiro–Wilk test is known not to work well in samples with many identical values and Jarque-Bera is bad for small samples as ours.The best test we can use seems to be Anderson-Darling.

##Test d'Anderson-Darling

```
ad.test(residuals(celec.lm4))
```

```
##
##  Anderson-Darling normality test
##
## data:  residuals(celec.lm4)
```

```
## A = 0.26891, p-value = 0.6187
```

The Anderson-Darling's test returns a p-value greater than 0.05 so we can consider that the residuals follows a gaussian distribution.

Vérification de l'hétéroscédasticité

To test for heteroscedasticity, we can choose between several tests. Since the Goldfeld-Quandt test is not very robust to specification errors and the White test is certainly more general and can detect a wider range of forms of heteroscedasticity, but cannot be used for small samples, we decide to use the Breusch-Pagan test. The latter is designed to detect only linear forms of heteroscedasticity, which could be our case.

##Test de Breusch-Pagan

```
bptest(celec.lm4)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  celec.lm4
## BP = 2.8577, df = 3, p-value = 0.4141
```

Here the test returns a p-value greater than 0.05 so we don't reject the null hypothesis and we assume homoscedasticity.

Test d'autocorrélation

##Test de Breusch-Godfrey

```
bgtest(celec.lm4)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  celec.lm4
## LM test = 1.6596, df = 1, p-value = 0.1977
```

##Etude de la multicolinéarité

```
vif(celec.lm4)
```

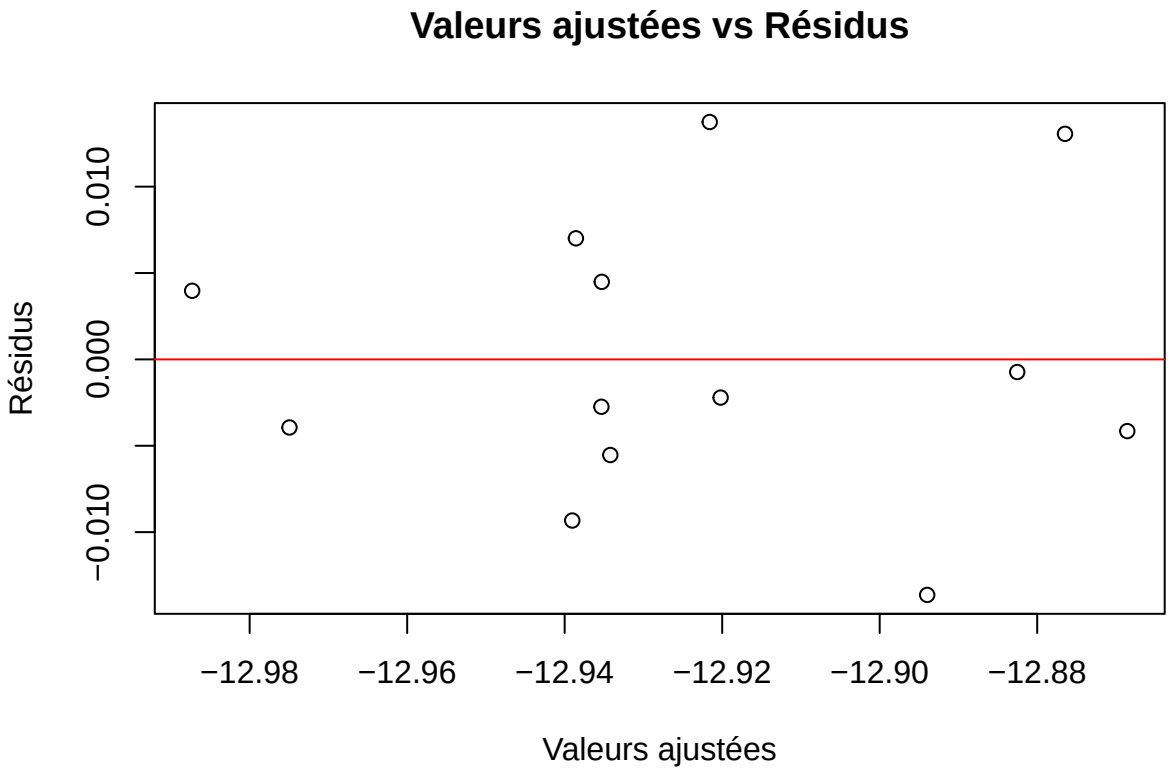
```
##     log(PIB2015_pc) log(Pelec_euro2015)                 IRC
##            2.046146            2.408974            1.259673
```

Pas de multicolinéarité

#Diagnostic visuel

##Graphique des valeurs ajustées vs résidus

```
plot(fitted(celec.lm4), residuals(celec.lm4), main="Valeurs ajustées vs Résidus", xlab="Valeurs ajustées
abline(h=0, col="red")
```



##QQ-plot des résidus

```
qqnorm(residuals(celec.lm4))
qqline(residuals(celec.lm4), col="red")
```