# R Notebook

## Contents

# 1 Paramétrage de la session

```
setwd('/home/julien1/Documents/BenoitBankMondial')
```

Option de mise en cache des chunks

## 1.1 Chargement des librairies

```
library(foreach)
library( )
```

```
## Warning in library(): library '/usr/lib/R/site-library' contains no packages
```

```
library(openxlsx) #pour la lecture des fichiers excels
library(foreach)
library(leaps)  # pour  la sélection de modèles
library(magrittr)
#install.packages('kableExtra')
#library(kableExtra)
```

## 1.2 Chargement des données

```
X <- read.xlsx("BDD_DEF_10mars2021++_2.xlsx", colNames=TRUE, rowNames = F, sheet=2)
X.all <- read.xlsx("BDD_DEF_10mars2021++_2.xlsx", colNames=TRUE, rowNames = F, sheet=1)
X.eur <- read.xlsx("BDD_DEF_10mars2021++_2.xlsx", colNames=TRUE, rowNames = F, sheet=6)
X.noneur <- read.xlsx("BDD_DEF_10mars2021++_2.xlsx", colNames=TRUE, rowNames = F, sheet=7)
X.grandes <- read.xlsx("BDD_DEF_10mars2021++_2.xlsx", colNames=TRUE, rowNames = F,
                       sheet=10)
X.petites <- read.xlsx("BDD_DEF_10mars2021++_2.xlsx", colNames=TRUE, rowNames = F,
                       sheet=11)



#list.files()
```

## 1.3 Fonctions

### 1.3.1 de bootstrap

Définition d'une fonction de bootstrap qui affiche les valeurs de coefficients, les intervalles de confiance bilatéraux à 95% des coefficients de la régression dans Estim_2.5 et Estim_97.5 , la pval de la reg unique, le pval_90 maximum avec son quantile à 90%. Affiche aussi le r2 ajusté minimum quantile à 5% et le r2 ajusté médian.

```
bootstrap <- function(f, data, boot=500){
cat('-----------------------------------------\n')
cat('Regression : ');print(f)
  sss <- summary(lm(formula = f, data=data))
  reg.coeff <- sss$coefficients[,c(1,4)]
  nvars <- length(attr(terms(f),"term.labels"))+1 # nb de variables dans le modèle, avec intercept
  foreach(i=1:boot,.combine=cbind)%do%{
  ind.boot <- sample(1:nrow(X), replace=T)
  reg <- lm(formula = f, data=X[ind.boot,])
  ss <- summary(reg)
  c(    ss$coefficients[,1],  # récupère la valeur des coeffs
        r2.adj = ss$adj.r.squared,  # récupère le r carré ajusté
        ss$coefficients[,c(4)] # récupère la p.value de chaque coef
    )
}-> pvals.full # tableau des 500 coef, pvaleurs et r2 adj de tous les coeffs et de chaque modèle
pvals_0.9 <- apply(
  pvals.full[(nrow(pvals.full)-nvars+1):nrow(pvals.full),],
  1,
  quantile, probs=c(0.9)
  ) # pvals de chaque coef de chaque variable
Estim_2.5 <- apply(
  pvals.full[1:nvars, ],
  1,
  quantile, probs=c(0.025)
  )
Estim_97.5 <- apply(
  pvals.full[1:nvars, ],
  1,
  quantile, probs=c(0.975)
  )

print(data.frame(
      Estim= round(reg.coeff[,1],2),
      pval = round(reg.coeff[,2],4),
      stars = star(reg.coeff[,2]),
      Estim_2.5 = round(Estim_2.5,2),
      Estim_97.5 = round(Estim_97.5,2),
      pval_90 = round(pvals_0.9,4),
      stars_90 = star(pvals_0.9)
    )
  )

 #print(reg.coeff)
  r2.adj.boot <- quantile(pvals.full[nvars+1, ], probs=c(0.05,0.5))
  cat(" r2 ajusté :\t",
      sss$adj.r.squared ,
      "\t quantile 5% =",
      r2.adj.boot[1],
      " \t quantile 50% =",
      r2.adj.boot[2],"\n")
}
```

### 1.3.2 de sélection de modèles

Fonction de sélection de variables dans les modèles

```
select.lm <- function(formula, data, nvmax=7 ){
  verbose=FALSE
  cat('--------------------------------------------\n')
  cat('Regression : '); print(formula)

  selection <- regsubsets(x=formula, data=data, nvmax=nvmax, method='forward') # dans leaps
  if(verbose)print(selection)
  summary(selection) ->  selection.summary
  #  selection.summary # pour afficher
  for(crit in c('bic','adjr2','cp')){
    cat('Selected by ',crit,'\t -------------------\n')
    num.var <- which.min(selection.summary[[crit]]) # min bic et cp
    # mais max pour le r2 ajusté
        if(crit=='adjr2')num.var <- which.max(selection.summary[[crit]])
    #coef(selection, num.var) # print names of selected vars and their coef
    xvars <-  paste(names(coef(selection,num.var))[-1],collapse='+')
    if(verbose)print(xvars)
    reg.formula <- formula(paste(y,'~',xvars))
    if(verbose)print(reg.formula)

    # Re-adjust the linear model
    reg.leaps <- lm(reg.formula, data=data)
    # and print the informations
    if(verbose)print(summary(reg.leaps))

    print(cbind(
        summary(reg.leaps)$coefficients[,c(1,4)]%>%round(4) ,
        star=star(summary(reg.leaps)$coefficients[,c(4)])
        ), quote=FALSE
    )
    cat('adj.r.squared = ',broom::glance(reg.leaps)[['adj.r.squared']],'     p.value = ',broom::glance(
  }
}
```

```
# A VERIFIER
 #select.lm(CDP ~ CEF + EPI + Taille + Croiss + Année+EUR, data=X.eur, nvmax=7)
```

# 2 Stats descriptives (+ outils Antoine)

## 2.1 Paramètres statistiques

```
k=8
library(foreach)
#summary(X[,-(1:6)])
foreach(k=7:ncol(X), .combine=rbind)%do%{
  c(colnames(X)[k],
     as.vector(round(summary(X[,k]),2))[1:6],
     round(sd(X[,k], na.rm=T),2) )

} -> stat.summary
ncol(stat.summary)
```

```
## [1] 8
```

```
colnames(stat.summary) <- c('Variable','Min','1st Q', 'Median', 'Mean', '3rd Q','Max','SD')
rownames(stat.summary) <- NULL
as.data.frame(stat.summary)
```

```
##    Variable   Min 1st Q Median  Mean 3rd Q   Max   SD
## 1       FEF  9.03  9.94  10.24 10.16 10.42 10.85 0.37
## 2       CEF  8.51  9.41   9.77  9.67    10  10.5 0.43
## 3       CDP     0     4      6  5.24     7     8 2.45
## 4       GRI     0     1      2  4.65     9    18 4.69
## 5        IV     2     8     13 11.54    15    22  5.1
## 6      ERSE     0     0      1  1.08     2     5 1.21
## 7     ERSEc     0     0      2  2.59     4    11 2.35
## 8       ECL     0     0      0  0.63     1     8 1.23
## 9      ECLc     0  0.75    1.5  2.96     4    16 3.37
## 10      COT     0     1      2  1.97     3     4 0.97
## 11      EUR     0     0      0  0.49     1     1  0.5
## 12      EPI 50.74 72.18  80.59 78.54 85.06 88.91 9.54
## 13   Taille 11.64 11.91  12.16 12.13 12.35 12.57 0.25
## 14 Taille.1     0     0    0.5   0.5     1     1  0.5
## 15   Croiss -0.36 -0.02   0.03  0.02  0.06  0.32 0.07
## 16    Renta -1.32  0.27   0.48  0.53  0.87  1.31 0.42
```

# 3 Tests Wilcoxon

## 3.1 Loading package of the article on github

```
library(devtools)
#devtools::install_github('Julien-Bousquet/Benoit-Jamet-BankInvestmentOnCarbon-2022', force=TRUE) # ins
library(monpackage) #load package of the github
```

## 3.2 Bootstraped Wilcoxon tests

```
set.seed(1)
names(X)
```

```
##  [1] "X1"       "Année"    "2016"     "2017"     "2018"     "2019"
##  [7] "FEF"      "CEF"      "CDP"      "GRI"      "IV"       "ERSE"
## [13] "ERSEc"    "ECL"      "ECLc"     "COT"      "EUR"      "EPI"
## [19] "Taille"   "Taille.1" "Croiss"   "Renta"
```

```
VARS <- c( 'FEF','CEF','CDP', 'GRI','ECLc', 'EPI', 'Taille', 'Croiss', 'Renta','ERSEc')
b <- 500 # Nombre de ré-échantillonnages dans le bootstrap (0=pas de bootstrap)
p <- 0

GROUPS <- c('FEF','CEF','CDP','ECLc','GRI','ERSEc','Taille','EUR','EPI')

library(foreach)
foreach(g = GROUPS, .combine=rbind, packages='foreach')%do%{
  wilcoxon.cut.test(x=VARS, group=g, bootstrap=b, prop=p, data=X)
}-> AnalyseWilcoxonComplete
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered

## p-value is in [ -2.136141e-24 ;  -1.784627e-24 ]
##  with quantile 50%= 1.800924e-24
##  and quantile 95%= 1.877389e-24
## p-value is in [ -1.099004e-17 ;  -3.402509e-24 ]
##  with quantile 50%= 8.605133e-22
##  and quantile 95%= 1.085108e-19

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## p-value is in [ -0.7824077 ;  0 ]
##  with quantile 50%= 0.001587892
##  and quantile 95%= 0.1652361
## p-value is in [ -0.9748608 ;  0.9761986 ]
##  with quantile 50%= 0.03397214
##  and quantile 95%= 0.7115634
## p-value is in [ -0.9865354 ;  1 ]
##  with quantile 50%= 0.04761028
##  and quantile 95%= 0.7578511
## p-value is in [ -0.8889091 ;  0.603664 ]
##  with quantile 50%= 0.003435192
##  and quantile 95%= 0.2105444
## p-value is in [ -0.9468213 ;  0.4126687 ]
##  with quantile 50%= 0.004659604
##  and quantile 95%= 0.2703426
## p-value is in [ -0.988165 ;  0.9898765 ]
##  with quantile 50%= 0.01107189
##  and quantile 95%= 0.7982568
```

```
AnalyseWilcoxonComplete
```

```
##      tested group.by   p.value.50   p.value.95       p.value star
## 50%     FEF      FEF 1.800924e-24 1.877389e-24 -1.877389e-24  ***
## 50%1    CEF      FEF 8.605133e-22 1.085108e-19 -1.084887e-19  ***
## 50%2    CDP      FEF 1.587892e-03 1.652361e-01 -1.651161e-01
## 50%3    GRI      FEF 3.397214e-02 7.115634e-01 -7.114578e-01
## 50%4   ECLc      FEF 4.761028e-02 7.578511e-01  7.574885e-01
## 50%5    EPI      FEF 6.163585e-02 7.748011e-01  7.747231e-01
## 50%6 Taille      FEF 3.435192e-03 2.105444e-01 -2.103916e-01
## 50%7 Croiss      FEF 4.217319e-01 9.290105e-01 -9.285967e-01
## 50%8  Renta      FEF 4.659604e-03 2.703426e-01 -2.701605e-01
## 50%9  ERSEc      FEF 1.107189e-02 7.982568e-01 -7.977455e-01
```

```
write.csv(AnalyseWilcoxonComplete,
          file='AnalyseWilcoxonComplete.csv',
          row.names = FALSE)

foreach(g = GROUPS, .combine=rbind)%do%{
  wilcoxon.cut.test(x=VARS, group=g, bootstrap=b, prop=0.2, data=X)
}-> AnalyseWilcoxonComplete
```

```
## p-value is in [ -1.569721e-19 ;  -1.537554e-19 ]
##  with quantile 50%= 1.561255e-19
##  and quantile 95%= 1.567381e-19
## p-value is in [ -3.333634e-18 ;  -1.543591e-19 ]
##  with quantile 50%= 2.052587e-19
##  and quantile 95%= 6.075783e-19
```

```
## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.
```

```
## p-value is in [ -1 ;  0.6684035 ]
##  with quantile 50%= 0.01985137
##  and quantile 95%= 0.6099723
## p-value is in [ -0.5744595 ;  0.9757241 ]
##  with quantile 50%= 0.002648983
##  and quantile 95%= 0.5235668
## p-value is in [ -1 ;  0.661164 ]
##  with quantile 50%= 0.01569452
##  and quantile 95%= 0.589419
## p-value is in [ -0.2738877 ;  0.1617838 ]
##  with quantile 50%= 0.0002361018
##  and quantile 95%= 0.04221555
## p-value is in [ -0.6408882 ;  0.2673025 ]
##  with quantile 50%= 0.0003543759
##  and quantile 95%= 0.06384111
## p-value is in [ -0.9951592 ;  0.9636759 ]
##  with quantile 50%= 0
##  and quantile 95%= 0.7936062
## p-value is in [ -6.264688e-18 ;  -1.540138e-19 ]
##  with quantile 50%= 2.417176e-19
##  and quantile 95%= 9.312642e-19
## p-value is in [ -1.569428e-19 ;  -1.540426e-19 ]
##  with quantile 50%= 1.561546e-19
##  and quantile 95%= 1.567089e-19

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## p-value is in [ -0.7851391 ;  0.9877866 ]
##  with quantile 50%= 0.001366678
##  and quantile 95%= 0.8003257
## p-value is in [ -0.9832623 ;  0.4722232 ]
##  with quantile 50%= 0.01121141
##  and quantile 95%= 0.4747668
## p-value is in [ -0.1824184 ;  -2.159677e-10 ]
##  with quantile 50%= 5.653717e-05
##  and quantile 95%= 0.01878219
## p-value is in [ -0.08284694 ;  -4.94739e-12 ]
##  with quantile 50%= 3.540096e-05
##  and quantile 95%= 0.01067034
## p-value is in [ -0.9975775 ;  0.9782415 ]
##  with quantile 50%= 0.005656867
##  and quantile 95%= 0.6184863

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## p-value is in [ -7.528216e-19 ;  -1.780815e-20 ]
##  with quantile 50%= 1.184985e-19
##  and quantile 95%= 2.477816e-19

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.
## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## p-value is in [ -0.5571468 ;  0.667068 ]
##  with quantile 50%= 0.001983048
##  and quantile 95%= 0.1846208

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## p-value is in [ -0.7980848 ;  1 ]
##  with quantile 50%= 0.005195436
##  and quantile 95%= 0.2369678

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## p-value is in [ -0.9973353 ;  0.9973071 ]
##  with quantile 50%= 0
##  and quantile 95%= 0.7779006
## p-value is in [ -4.618e-20 ;  -2.455991e-21 ]
##  with quantile 50%= 2.69911e-20
##  and quantile 95%= 3.619981e-20
## p-value is in [ -0.99043 ;  0.8856296 ]
##  with quantile 50%= 0.01368067
##  and quantile 95%= 0.5080554
## p-value is in [ -0.3009727 ;  -1.719622e-12 ]
##  with quantile 50%= 7.372723e-05
##  and quantile 95%= 0.01425586
## p-value is in [ -0.03185676 ;  0 ]
##  with quantile 50%= 8.713685e-07
##  and quantile 95%= 0.0005240306

## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.

## p-value is in [ -0.932819 ;  0.9602202 ]
##  with quantile 50%= 0
##  and quantile 95%= 0.7657781
## p-value is in [ -7.767132e-20 ;  -4.453921e-21 ]
##  with quantile 50%= 3.230889e-20
##  and quantile 95%= 5.771433e-20
## p-value is in [ -1 ;  0.7399242 ]
##  with quantile 50%= 0
##  and quantile 95%= 0.869293
## p-value is in [ -0.9732935 ;  0.8724447 ]
##  with quantile 50%= 0.009490918
##  and quantile 95%= 0.3533363
## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.
```

```
## p-value is in [ -0.3329054 ;  0.992002 ]
##  with quantile 50%= 0.0006535427
##  and quantile 95%= 0.0986979
## p-value is in [ -0.1107581 ;  -2.313438e-13 ]
##  with quantile 50%= 6.237316e-07
##  and quantile 95%= 0.0004812098
## p-value is in [ -0.9641457 ;  0.9976076 ]
##  with quantile 50%= 0.02480425
##  and quantile 95%= 0.5046973
## p-value is in [ -6.033783e-20 ;  -2.29793e-22 ]
##  with quantile 50%= 2.212472e-20
##  and quantile 95%= 4.660801e-20
## p-value is in [ -0.02002517 ;  0.009792845 ]
##  with quantile 50%= 1.943216e-06
##  and quantile 95%= 0.0005170075
## p-value is in [ -0.006880268 ;  0.02417127 ]
##  with quantile 50%= 9.02919e-08
##  and quantile 95%= 0.000121242
```

```
## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.
```

```
## p-value is in [ -0.9678188 ;  0.5155916 ]
##  with quantile 50%= 0.003232032
##  and quantile 95%= 0.4455284
## p-value is in [ -0.01871987 ;  0 ]
##  with quantile 50%= 3.378314e-06
##  and quantile 95%= 0.0008165776
## p-value is in [ -0.9092269 ;  0.3996968 ]
##  with quantile 50%= 0.0001371507
##  and quantile 95%= 0.05797209
## p-value is in [ -1.571477e-19 ;  -1.542439e-19 ]
##  with quantile 50%= 1.561546e-19
##  and quantile 95%= 1.567089e-19
## p-value is in [ -0.625953 ;  0.9618429 ]
##  with quantile 50%= 0.002405528
##  and quantile 95%= 0.272461
## p-value is in [ -0.9782398 ;  0.9584765 ]
##  with quantile 50%= 0.01063757
##  and quantile 95%= 0.7384844
##
## Please convert groups in reals. Or use bootstrap=0.
## p-value is in [ 4.963394e-19 ;  6.611539e-07 ]
##  with quantile 50%= 1.909668e-13
##  and quantile 95%= 5.354863e-10
##
## Please convert groups in reals. Or use bootstrap=0.
## p-value is in [ 2.17298e-20 ;  1.370151e-05 ]
##  with quantile 50%= 5.657827e-14
##  and quantile 95%= 1.385812e-10
```

```
## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.
```

```
##
## Please convert groups in reals. Or use bootstrap=0.
## p-value is in [ -0.4358008 ;  0.9567297 ]
##  with quantile 50%= 0
##  and quantile 95%= 0.2843814
##
## Please convert groups in reals. Or use bootstrap=0.
##
## Please convert groups in reals. Or use bootstrap=0.
## p-value is in [ -0.7311738 ;  0 ]
##  with quantile 50%= 0.006781334
##  and quantile 95%= 0.3062536
##
## Please convert groups in reals. Or use bootstrap=0.
## p-value is in [ -9.745805e-05 ;  0.0002194797 ]
##  with quantile 50%= 1.844917e-09
##  and quantile 95%= 1.641189e-06
##
## Please convert groups in reals. Or use bootstrap=0.
## p-value is in [ -0.1527391 ;  0.1356284 ]
##  with quantile 50%= 0.0002479528
##  and quantile 95%= 0.04274986
##
## Please convert groups in reals. Or use bootstrap=0.
## p-value is in [ 2.564105e-09 ;  0.6914656 ]
##  with quantile 50%= 0.0002333051
##  and quantile 95%= 0.0265197
##
## Please convert groups in reals. Or use bootstrap=0.
## p-value is in [ 5.380267e-23 ;  1.203552e-09 ]
##  with quantile 50%= 1.330245e-17
##  and quantile 95%= 3.210861e-14
##
## Please convert groups in reals. Or use bootstrap=0.
## p-value is in [ -0.04270856 ;  0 ]
##  with quantile 50%= 3.062293e-05
##  and quantile 95%= 0.004463312
```

```
## Warning in wilcoxon.cut.test(x = i, group = group, data = data, prop = prop, :
## Missing data removed.
```

```
## p-value is in [ -0.4069763 ;  0 ]
##  with quantile 50%= 0.0005479639
##  and quantile 95%= 0.05802174
## p-value is in [ -0.9878503 ;  0.8123853 ]
##  with quantile 50%= 0.00133547
##  and quantile 95%= 0.2330653
## p-value is in [ -1.342385e-19 ;  -7.944237e-20 ]
##  with quantile 50%= 1.160588e-19
##  and quantile 95%= 1.2642e-19
## p-value is in [ -0.9142935 ;  0.7332504 ]
##  with quantile 50%= 0.0006834687
##  and quantile 95%= 0.1292318
## p-value is in [ -0.9952295 ;  0.957084 ]
##  with quantile 50%= 0.03328887
##  and quantile 95%= 0.6931178
## p-value is in [ 1.645663e-15 ;  0.2080863 ]
##  with quantile 50%= 3.782261e-07
##  and quantile 95%= 0.0009076896
## p-value is in [ -0.6313064 ;  0.7591189 ]
##  with quantile 50%= 0.0005183863
##  and quantile 95%= 0.1238266
```

```
AnalyseWilcoxonComplete
```

```
##       tested group.by   p.value.50   p.value.95       p.value star
## 50%      FEF      FEF 1.561255e-19 1.567381e-19 -1.567381e-19  ***
## 50%1     CEF      FEF 2.052587e-19 6.075783e-19 -6.058060e-19  ***
## 50%2     CDP      FEF 1.985137e-02 6.099723e-01 -6.096492e-01
## 50%3     GRI      FEF 8.736139e-02 7.899421e-01 -7.895529e-01
## 50%4    ECLc      FEF 2.648983e-03 5.235668e-01  5.228658e-01
## 50%5     EPI      FEF 1.569452e-02 5.894190e-01 -5.892139e-01
## 50%6  Taille      FEF 2.361018e-04 4.221555e-02 -4.205946e-02    *
## 50%7  Croiss      FEF 2.364581e-01 9.240263e-01  9.237889e-01
## 50%8   Renta      FEF 3.543759e-04 6.384111e-02 -6.377771e-02    .
## 50%9   ERSEc      FEF 0.000000e+00 7.936062e-01 -7.935525e-01
## 50%10    FEF      CEF 2.417176e-19 9.312642e-19 -9.312227e-19  ***
## 50%11    CEF      CEF 1.561546e-19 1.567089e-19 -1.567089e-19  ***
## 50%21    CDP      CEF 8.655603e-02 8.134423e-01 -8.130886e-01
## 50%31    GRI      CEF 1.343553e-01 8.590319e-01 -8.586983e-01
## 50%41   ECLc      CEF 1.366678e-03 8.003257e-01  8.002241e-01
## 50%51    EPI      CEF 1.121141e-02 4.747668e-01 -4.739397e-01
## 50%61 Taille      CEF 5.653717e-05 1.878219e-02 -1.878201e-02    *
## 50%71 Croiss      CEF 2.017602e-01 8.534232e-01 -8.529535e-01
## 50%81  Renta      CEF 3.540096e-05 1.067034e-02 -1.066543e-02    *
## 50%91  ERSEc      CEF 5.656867e-03 6.184863e-01  6.168073e-01
## 50%12    FEF      CDP 9.141345e-02 7.877933e-01 -7.872926e-01
## 50%13    CEF      CDP 6.433334e-02 7.184436e-01  7.182011e-01
## 50%22    CDP      CDP 1.184985e-19 2.477816e-19 -2.474958e-19  ***
## 50%32    GRI      CDP 1.854165e-01 9.110454e-01  9.106257e-01
## 50%42   ECLc      CDP 2.466506e-01 9.394127e-01 -9.394082e-01
## 50%52    EPI      CDP 1.983048e-03 1.846208e-01 -1.845155e-01
## 50%62 Taille      CDP 5.802212e-02 7.574319e-01  7.574319e-01
## 50%72 Croiss      CDP 1.033472e-01 8.580861e-01  8.580859e-01
## 50%82  Renta      CDP 2.398009e-01 8.991983e-01 -8.990693e-01
## 50%92  ERSEc      CDP 5.195436e-03 2.369678e-01 -2.367900e-01
## 50%14    FEF     ECLc 1.635855e-01 8.955892e-01 -8.953531e-01
## 50%15    CEF     ECLc 1.110716e-01 8.393781e-01 -8.389102e-01
## 50%23    CDP     ECLc 0.000000e+00 7.779006e-01  7.778293e-01
## 50%33    GRI     ECLc 2.825270e-01 9.376802e-01  9.374348e-01
## 50%43   ECLc     ECLc 2.699110e-20 3.619981e-20 -3.619583e-20  ***
## 50%53    EPI     ECLc 1.368067e-02 5.080554e-01  5.079705e-01
## 50%63 Taille     ECLc 7.372723e-05 1.425586e-02 -1.423176e-02    *
## 50%73 Croiss     ECLc 8.453268e-02 8.017255e-01 -8.017253e-01
## 50%83  Renta     ECLc 1.302718e-01 8.295344e-01  8.295341e-01
## 50%93  ERSEc     ECLc 8.713685e-07 5.240306e-04 -5.226379e-04  ***
## 50%16    FEF      GRI 2.473073e-01 9.475581e-01 -9.475580e-01
## 50%17    CEF      GRI 4.584726e-01 9.428025e-01  9.428025e-01
## 50%24    CDP      GRI 0.000000e+00 7.657781e-01  7.654398e-01
## 50%34    GRI      GRI 3.230889e-20 5.771433e-20 -5.770982e-20  ***
## 50%44    ECLc      GRI 0.000000e+00 8.692930e-01 -8.685286e-01
## 50%54     EPI      GRI 4.242536e-01 9.191158e-01 -9.188801e-01
## 50%64  Taille      GRI 4.265019e-01 9.428023e-01  9.428023e-01
## 50%74  Croiss      GRI 4.247538e-01 9.382821e-01  9.380442e-01
## 50%84   Renta      GRI 3.193686e-01 9.287628e-01  9.285252e-01
## 50%94   ERSEc      GRI 9.490918e-03 3.533363e-01  3.532759e-01
## 50%18     FEF    ERSEc 5.030836e-01 9.428022e-01  9.428021e-01
## 50%19     CEF    ERSEc 5.320973e-01 9.523240e-01  9.523240e-01
## 50%25     CDP    ERSEc 6.535427e-04 9.869790e-02 -9.853659e-02    .
## 50%35     GRI    ERSEc 1.693205e-01 8.852610e-01  8.849051e-01
## 50%45    ECLc    ERSEc 6.237316e-07 4.812098e-04 -4.810229e-04  ***
## 50%55     EPI    ERSEc 2.480425e-02 5.046973e-01  5.040227e-01
## 50%65  Taille    ERSEc 4.992600e-01 9.332902e-01 -9.332901e-01
## 50%75  Croiss    ERSEc 5.030759e-01 9.332894e-01  9.332894e-01
## 50%85   Renta    ERSEc 5.016092e-02 6.909609e-01  6.908507e-01
## 50%95   ERSEc    ERSEc 2.212472e-20 4.660801e-20 -4.659232e-20  ***
## 50%20     FEF   Taille 1.943216e-06 5.170075e-04 -5.133615e-04  ***
## 50%110    CEF   Taille 9.029190e-08 1.212420e-04 -1.207847e-04  ***
## 50%26     CDP   Taille 3.232032e-03 4.455284e-01 -4.432219e-01
## 50%36     GRI   Taille 3.530816e-01 9.373910e-01 -9.373856e-01
## 50%46    ECLc   Taille 3.378314e-06 8.165776e-04 -8.063044e-04  ***
## 50%56     EPI   Taille 1.371507e-04 5.797209e-02  5.785087e-02    .
## 50%66  Taille   Taille 1.561546e-19 1.567089e-19 -1.567089e-19  ***
## 50%76  Croiss   Taille 5.494907e-02 7.832825e-01  7.832822e-01
## 50%86   Renta   Taille 2.405528e-03 2.724610e-01  2.724589e-01
## 50%96   ERSEc   Taille 1.063757e-02 7.384844e-01 -7.382356e-01
## 50%27     FEF      EUR 1.909668e-13 5.354863e-10  5.320457e-10  ***
## 50%111    CEF      EUR 5.657827e-14 1.385812e-10  1.369735e-10  ***
## 50%28     CDP      EUR 0.000000e+00 2.843814e-01  2.828360e-01
## 50%37     GRI      EUR 2.094999e-01 9.381757e-01  9.381627e-01
## 50%47    ECLc      EUR 6.781334e-03 3.062536e-01 -3.055345e-01
## 50%57     EPI      EUR 1.844917e-09 1.641189e-06 -1.638368e-06  ***
## 50%67  Taille      EUR 2.479528e-04 4.274986e-02  4.251343e-02    *
## 50%77  Croiss      EUR 2.333051e-04 2.651970e-02  2.647992e-02    *
## 50%87   Renta      EUR 1.330245e-17 3.210861e-14  3.125045e-14  ***
## 50%97   ERSEc      EUR 3.062293e-05 4.463312e-03 -4.440874e-03   **
## 50%29     FEF      EPI 2.225615e-01 9.287773e-01 -9.285399e-01
## 50%112    CEF      EPI 1.644993e-01 8.627981e-01  8.623270e-01
## 50%210    CDP      EPI 5.479639e-04 5.802174e-02 -5.800791e-02    .
## 50%38     GRI      EPI 2.742039e-01 9.112919e-01  9.111731e-01
## 50%48    ECLc      EPI 1.335470e-03 2.330653e-01  2.328466e-01
## 50%58     EPI      EPI 1.160588e-19 1.264200e-19 -1.264200e-19  ***
## 50%68  Taille      EPI 6.834687e-04 1.292318e-01  1.288461e-01
## 50%78  Croiss      EPI 3.328887e-02 6.931178e-01 -6.931176e-01
## 50%88   Renta      EPI 3.782261e-07 9.076896e-04  9.019749e-04  ***
## 50%98   ERSEc      EPI 5.183863e-04 1.238266e-01  1.236165e-01
```

```
write.csv(AnalyseWilcoxonComplete,
          file='AnalyseWilcoxonComplete20pc.csv',
          row.names = FALSE)
```

# 4 Corrélations

FEF; CEF; CDP; GRI; ERSEc; ECLc; EPI; Taille; Croiss; Renta

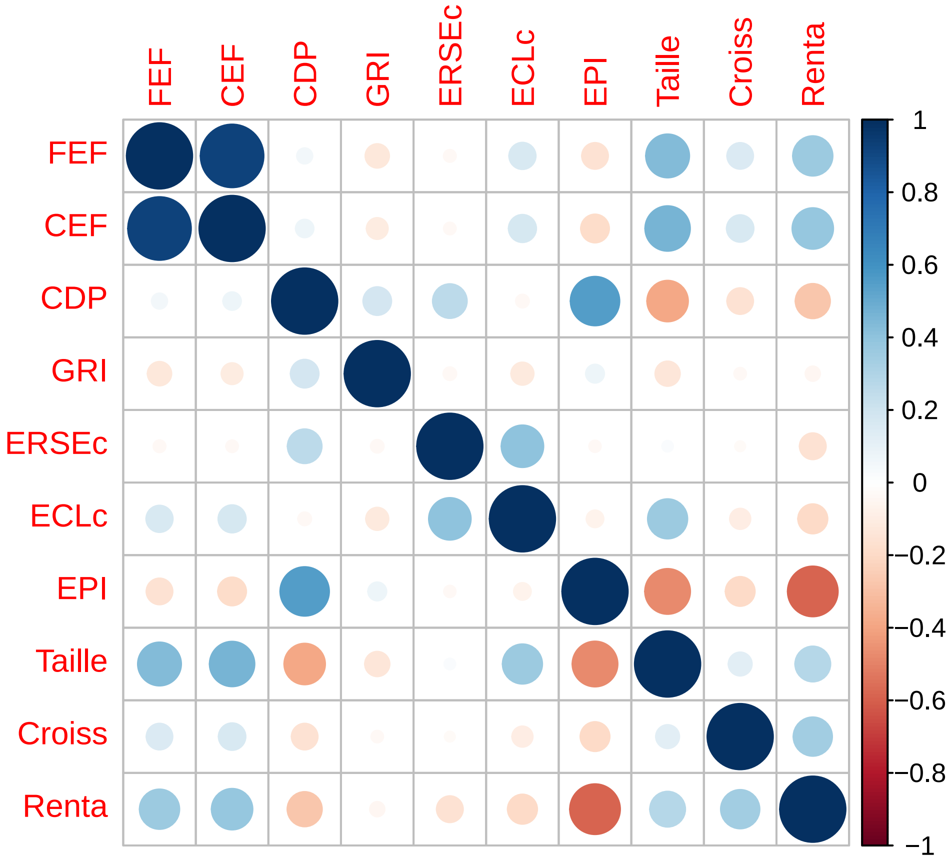
```r
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
#X[,c('FEF','CEF','CDP','GRI','ERSEc','ECLc','EPI','Taille','Croiss','Renta')]

M <- cor(X[,c('FEF','CEF','CDP','GRI','ERSEc','ECLc','EPI','Taille','Croiss','Renta')], use='pairwise.c

corrplot(M, )
```



```r
devtools::install_github("Antoine-Masse/KefiR") # pour corrigraph
```

```
## Skipping install of 'KefiR' from a github remote, the SHA1 (2c49343e) has not changed since last ins
##   Use `force = TRUE` to force installation
```

```r
library(KefiR)
# Fonction en cours de développement pour Rstudio :
# fonctionne dans Rgui uniquement
# corrigraph(X[,c('FEF','CEF','CDP','GRI','ERSEc','ECLc','EPI','Taille','Croiss','Renta')])
```

## 5 Leaps : écremage de modèle

Ecrémage automatique pour toutes les variables, et ajustement du modèle le plus pertinent avec affichage résumé. Le mode de sélection du modèle n'impacte pas.

```r
for(y in c('CDP','ECLc','GRI','ERSEc')){
  f <- formula(paste(y,'~FEF*CEF+EPI+COT+Taille+Croiss+Renta+EUR+Année'))
```

```
 select.lm(f, data=X, nvmax=7)
}
```

```
## --------------------------------------------
## Regression : CDP ~ FEF * CEF + EPI + COT + Taille + Croiss + Renta + EUR +
##     Année
## Selected by  bic      --------------------
##             Estimate Pr(>|t|) star
## (Intercept) 358.7099 0        ***
## FEF         -37.0264 0        ***
## CEF         -34.3773 1e-04    ***
## EPI         0.1363   0        ***
## Taille      -2.3632  0.0029   **
## Croiss      -2.5158  0.2693
## Année       0.4077   0.011    *
## FEF:CEF     3.7194   0        ***
## adj.r.squared =  0.4794321      p.value =  1.630567e-16
##
## Selected by  adjr2    --------------------
##             Estimate Pr(>|t|) star
## (Intercept) 358.7099 0        ***
## FEF         -37.0264 0        ***
## CEF         -34.3773 1e-04    ***
## EPI         0.1363   0        ***
## Taille      -2.3632  0.0029   **
## Croiss      -2.5158  0.2693
## Année       0.4077   0.011    *
## FEF:CEF     3.7194   0        ***
## adj.r.squared =  0.4794321      p.value =  1.630567e-16
##
## Selected by  cp   --------------------
##             Estimate Pr(>|t|) star
## (Intercept) 358.7099 0        ***
## FEF         -37.0264 0        ***
## CEF         -34.3773 1e-04    ***
## EPI         0.1363   0        ***
## Taille      -2.3632  0.0029   **
## Croiss      -2.5158  0.2693
## Année       0.4077   0.011    *
## FEF:CEF     3.7194   0        ***
## adj.r.squared =  0.4794321      p.value =  1.630567e-16
##
## --------------------------------------------
## Regression : ECLc ~ FEF * CEF + EPI + COT + Taille + Croiss + Renta + EUR +
##     Année
## Selected by  bic      --------------------
##             Estimate Pr(>|t|) star
## (Intercept) -44.0417 0.0026   **
## EPI         -0.1163  0.001    ***
## Taille      2.4212   0.05     .
## EUR         5.63     0        ***
## FEF:CEF     0.2442   0        ***
## adj.r.squared =  0.3794102      p.value =  4.005267e-14
##
```

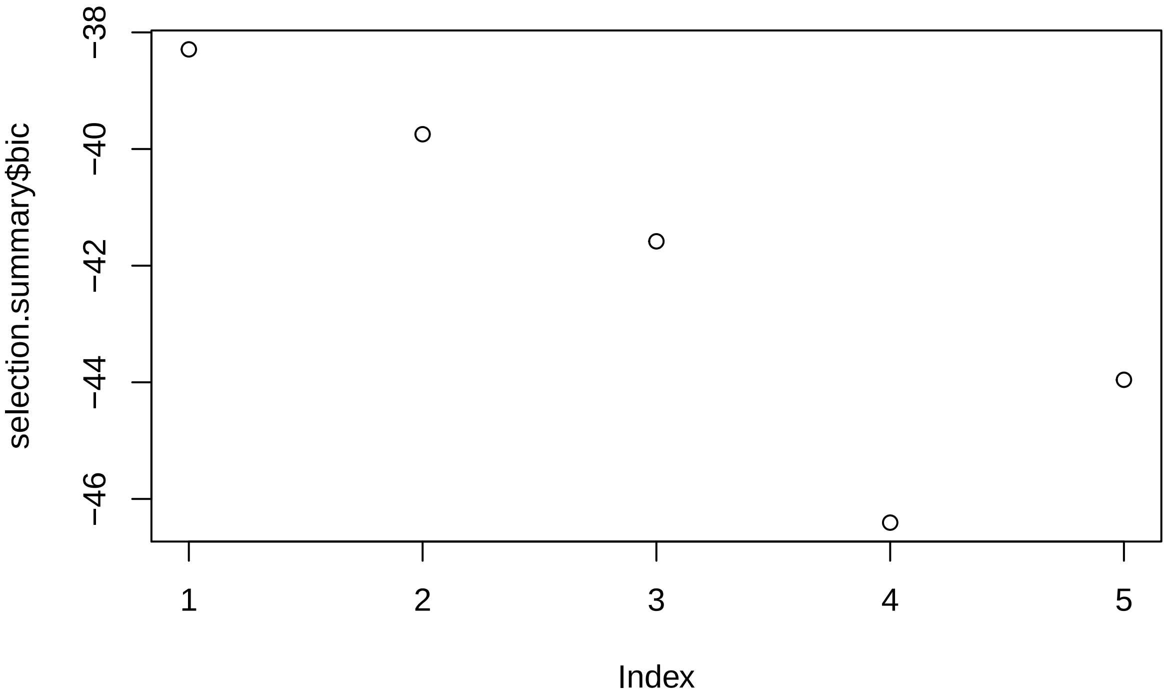
```
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -39.7523 0.0057   **
## EPI         -0.1427  1e-04    ***
## COT         -0.4617  0.0478   *
## Taille      2.2533   0.0632   .
## Renta       -1.4056  0.07     .
## EUR         5.204    0        ***
## FEF:CEF     0.2614   0        ***
## adj.r.squared =  0.4039998      p.value =  2.672721e-14
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -39.7523 0.0057   **
## EPI         -0.1427  1e-04    ***
## COT         -0.4617  0.0478   *
## Taille      2.2533   0.0632   .
## Renta       -1.4056  0.07     .
## EUR         5.204    0        ***
## FEF:CEF     0.2614   0        ***
## adj.r.squared =  0.4039998      p.value =  2.672721e-14
##
## --------------------------------------------
## Regression : GRI ~ FEF * CEF + EPI + COT + Taille + Croiss + Renta + EUR +
##     Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 36.5672  0.0598   .
## Taille      -2.6318  0.0998   .
## adj.r.squared =  0.01240338      p.value =  0.09978681
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 36.5672  0.0598   .
## Taille      -2.6318  0.0998   .
## adj.r.squared =  0.01240338      p.value =  0.09978681
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 36.5672  0.0598   .
## Taille      -2.6318  0.0998   .
## adj.r.squared =  0.01240338      p.value =  0.09978681
##
## --------------------------------------------
## Regression : ERSEc ~ FEF * CEF + EPI + COT + Taille + Croiss + Renta + EUR +
##     Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -35.3474 0        ***
## CEF         1.5441   2e-04    ***
## EUR         2.5427   0        ***
## Année       1.2439   0        ***
## adj.r.squared =  0.5371633      p.value =  2.801938e-23
##
```

```
## Selected by  adjr2   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 114.0535 0.098    .
## FEF         -14.9821 0.0318   *
## CEF         -14.0962 0.0518   .
## COT         -0.0998  0.482
## Renta       0.1788   0.6942
## EUR         2.7685   0        ***
## Année       1.2235   0        ***
## FEF:CEF     1.5693   0.0298   *
## adj.r.squared =  0.5419516      p.value =  3.866869e-21
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -35.3474 0        ***
## CEF         1.5441   2e-04    ***
## EUR         2.5427   0        ***
## Année       1.2439   0        ***
## adj.r.squared =  0.5371633      p.value =  2.801938e-23
```

```
selection <- regsubsets(CDP~FEF+CEF+EPI+Taille+Croiss+Renta+Année+EUR,
                        data=X, nvmax=5, method='forward')
summary(selection) ->  selection.summary
  selection.summary
```

```
## Subset selection object
## Call: regsubsets.formula(CDP ~ FEF + CEF + EPI + Taille + Croiss +
##     Renta + Année + EUR, data = X, nvmax = 5, method = "forward")
## 8 Variables  (and intercept)
##        Forced in Forced out
## FEF        FALSE      FALSE
## CEF        FALSE      FALSE
## EPI        FALSE      FALSE
## Taille     FALSE      FALSE
## Croiss     FALSE      FALSE
## Renta      FALSE      FALSE
## Année      FALSE      FALSE
## EUR        FALSE      FALSE
## 1 subsets of each size up to 5
## Selection Algorithm: forward
##          FEF CEF EPI Taille Croiss Renta Année EUR
## 1  ( 1 ) " " " " "*" " "    " "    " "   " "   " "
## 2  ( 1 ) " " " " "*" " "    " "    " "   "*"   " "
## 3  ( 1 ) " " "*" "*" " "    " "    " "   "*"   " "
## 4  ( 1 ) " " "*" "*" "*"    " "    " "   "*"   " "
## 5  ( 1 ) " " "*" "*" "*"    "*"    " "   "*"   " "
```

```
plot(selection.summary$bic)
```



```
coef(selection,4)
```

```
## (Intercept)         CEF         EPI      Taille       Année
##   3.3427957   1.6288761   0.1401063  -2.6002996   0.3791030
```

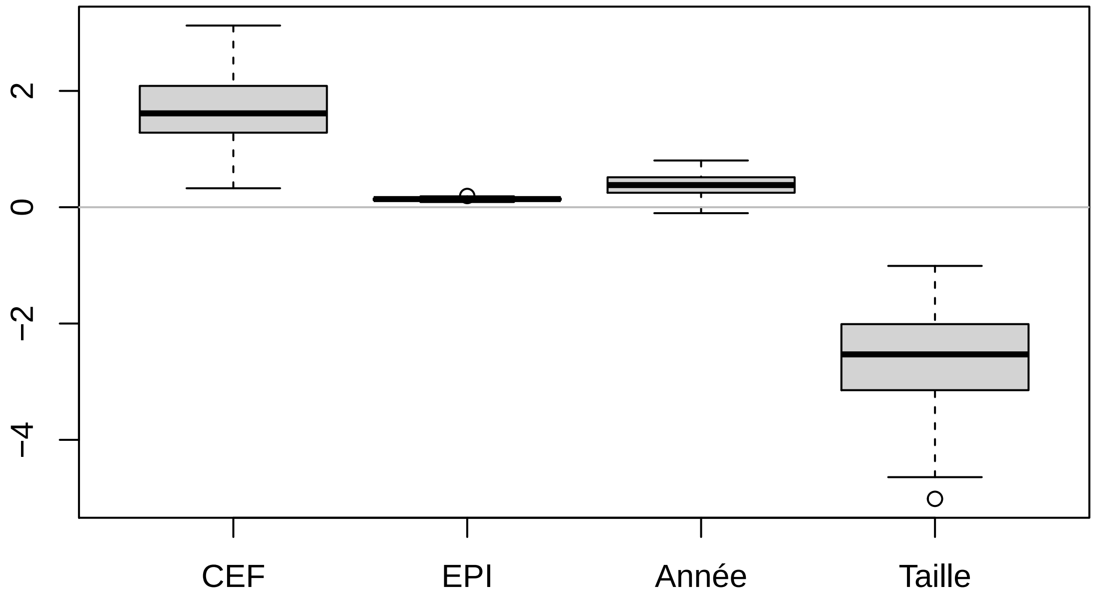
```
reg1 <- lm(CDP~CEF+EPI+Année+Taille, data=X)
summary(reg1)
```

```
##
## Call:
## lm(formula = CDP ~ CEF + EPI + Année + Taille, data = X)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.898 -1.181  0.377  1.294  3.631
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.34280   11.26028   0.297 0.767054
## CEF          1.62888    0.43817   3.717 0.000301 ***
## EPI          0.14011    0.02192   6.393 2.84e-09 ***
## Année        0.37910    0.16787   2.258 0.025632 *
## Taille      -2.60030    0.83534  -3.113 0.002290 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.9 on 127 degrees of freedom
##   (8 observations deleted due to missingness)
## Multiple R-squared:  0.4152, Adjusted R-squared:  0.3968
## F-statistic: 22.54 on 4 and 127 DF,  p-value: 4.38e-14
```

```
is(X)
```

```
## [1] "data.frame" "list"       "oldClass"   "vector"
```

```
set.seed(1)
foreach(i = 1:100, .combine=rbind)%do%{
  ind <- sample(1:nrow(X), replace=T)
  reg <- lm(CDP~CEF+EPI+Année+Taille, data=X[ind,])
  coef(reg)
} -> COEF.boot
boxplot(COEF.boot[,-1])
abline(h=0, col='grey')
```



```
reg$coefficients
```

```
## (Intercept)         CEF         EPI       Année      Taille
##   9.5187351   2.4654584   0.1238625   0.2393435  -3.4922123
```

```
names(summary(reg))
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"  "na.action"
```

```r
summary(reg)
```

```
##
## Call:
## lm(formula = CDP ~ CEF + EPI + Année + Taille, data = X[ind,
##     ])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.006 -1.010  0.343  1.426  3.740
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.51874   13.10313   0.726 0.468923
## CEF          2.46546    0.44835   5.499 2.06e-07 ***
## EPI          0.12386    0.02391   5.181 8.60e-07 ***
## Année        0.23934    0.18772   1.275 0.204677
## Taille      -3.49221    0.91311  -3.825 0.000206 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.995 on 125 degrees of freedom
##   (10 observations deleted due to missingness)
## Multiple R-squared:  0.4238, Adjusted R-squared:  0.4054
## F-statistic: 22.99 on 4 and 125 DF,  p-value: 2.974e-14
```

```r
summary(X)
```

```
##       X1                Année           2016           2017
##  Length:140         Min.   :16.00   Min.   :0.00   Min.   :0.00
##  Class :character   1st Qu.:16.75   1st Qu.:0.00   1st Qu.:0.00
##  Mode  :character   Median :17.50   Median :0.00   Median :0.00
##                     Mean   :17.50   Mean   :0.25   Mean   :0.25
##                     3rd Qu.:18.25   3rd Qu.:0.25   3rd Qu.:0.25
##                     Max.   :19.00   Max.   :1.00   Max.   :1.00
##
##       2018           2019           FEF              CEF
##  Min.   :0.00   Min.   :0.00   Min.   : 9.032   Min.   : 8.506
##  1st Qu.:0.00   1st Qu.:0.00   1st Qu.: 9.942   1st Qu.: 9.412
##  Median :0.00   Median :0.00   Median :10.245   Median : 9.766
##  Mean   :0.25   Mean   :0.25   Mean   :10.160   Mean   : 9.673
##  3rd Qu.:0.25   3rd Qu.:0.25   3rd Qu.:10.416   3rd Qu.: 9.996
##  Max.   :1.00   Max.   :1.00   Max.   :10.849   Max.   :10.500
##
##       CDP             GRI              IV              ERSE
##  Min.   :0.000   Min.   : 0.00   Min.   : 2.00   Min.   :0.000
##  1st Qu.:4.000   1st Qu.: 1.00   1st Qu.: 8.00   1st Qu.:0.000
##  Median :6.000   Median : 2.00   Median :13.00   Median :1.000
##  Mean   :5.242   Mean   : 4.65   Mean   :11.54   Mean   :1.079
##  3rd Qu.:7.000   3rd Qu.: 9.00   3rd Qu.:15.00   3rd Qu.:2.000
##  Max.   :8.000   Max.   :18.00   Max.   :22.00   Max.   :5.000
##  NA's   :8
##      ERSEc              ECL               ECLc              COT
##  Min.   : 0.000   Min.   :0.0000   Min.   : 0.000   Min.   :0.000
##  1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.: 0.750   1st Qu.:1.000  
##  Median : 2.000   Median :0.0000   Median : 1.500   Median :2.000  
##  Mean   : 2.593   Mean   :0.6286   Mean   : 2.964   Mean   :1.971  
##  3rd Qu.: 4.000   3rd Qu.:1.0000   3rd Qu.: 4.000   3rd Qu.:3.000  
##  Max.   :11.000   Max.   :8.0000   Max.   :16.000   Max.   :4.000  
##                                                                    
##       EUR              EPI            Taille         Taille.1  
##  Min.   :0.0000   Min.   :50.74   Min.   :11.64   Min.   :0.0  
##  1st Qu.:0.0000   1st Qu.:72.18   1st Qu.:11.91   1st Qu.:0.0  
##  Median :0.0000   Median :80.59   Median :12.16   Median :0.5  
##  Mean   :0.4857   Mean   :78.54   Mean   :12.13   Mean   :0.5  
##  3rd Qu.:1.0000   3rd Qu.:85.06   3rd Qu.:12.35   3rd Qu.:1.0  
##  Max.   :1.0000   Max.   :88.91   Max.   :12.57   Max.   :1.0  
##                                                                
##      Croiss             Renta        
##  Min.   :-0.35941   Min.   :-1.3200  
##  1st Qu.:-0.01801   1st Qu.: 0.2675  
##  Median : 0.02565   Median : 0.4850  
##  Mean   : 0.02194   Mean   : 0.5306  
##  3rd Qu.: 0.05527   3rd Qu.: 0.8725  
##  Max.   : 0.31630   Max.   : 1.3100  
## 
```

```
# On détermine les meilleures variables
pred <- paste(names(coef(selection,5))[-1], collapse="+")

coef(selection,1:3)
```

```
## [[1]]
## (Intercept)         EPI 
##  -5.7425359   0.1395829 
## 
## [[2]]
## (Intercept)         EPI       Année 
## -15.2033431   0.1621646   0.4390660 
## 
## [[3]]
## (Intercept)         CEF         EPI       Année 
## -26.7667276   1.0665414   0.1715396   0.4678261
```

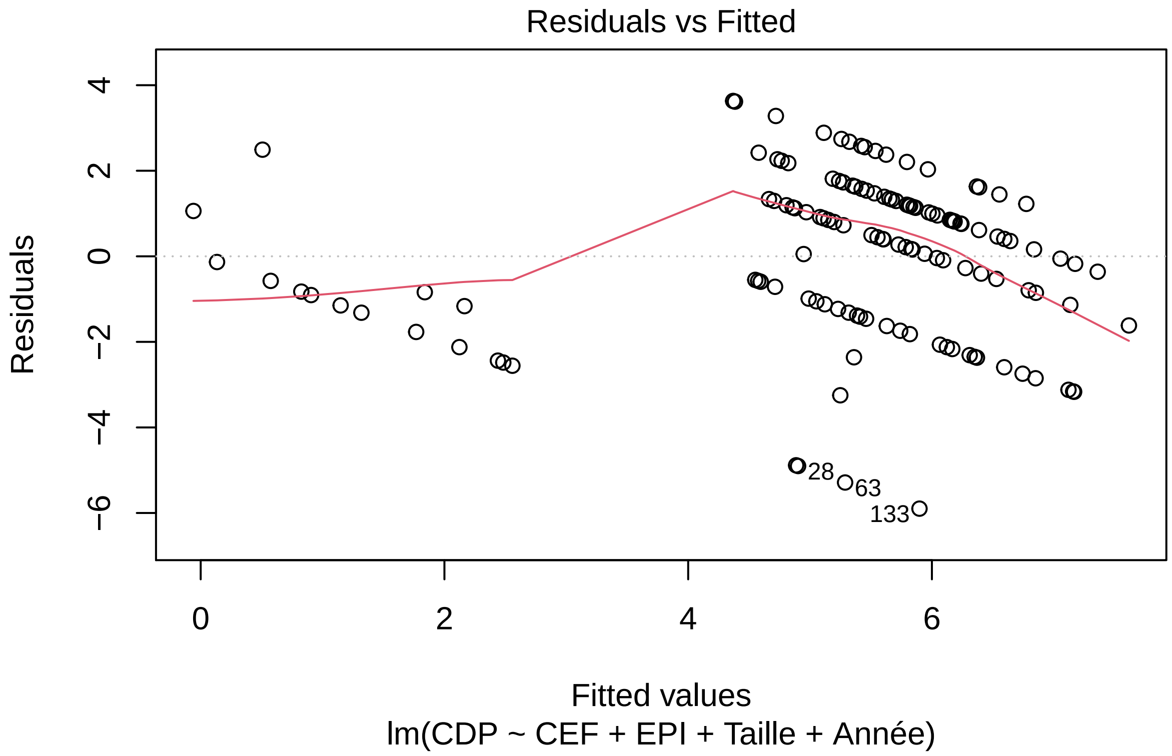
```
selection$which
```

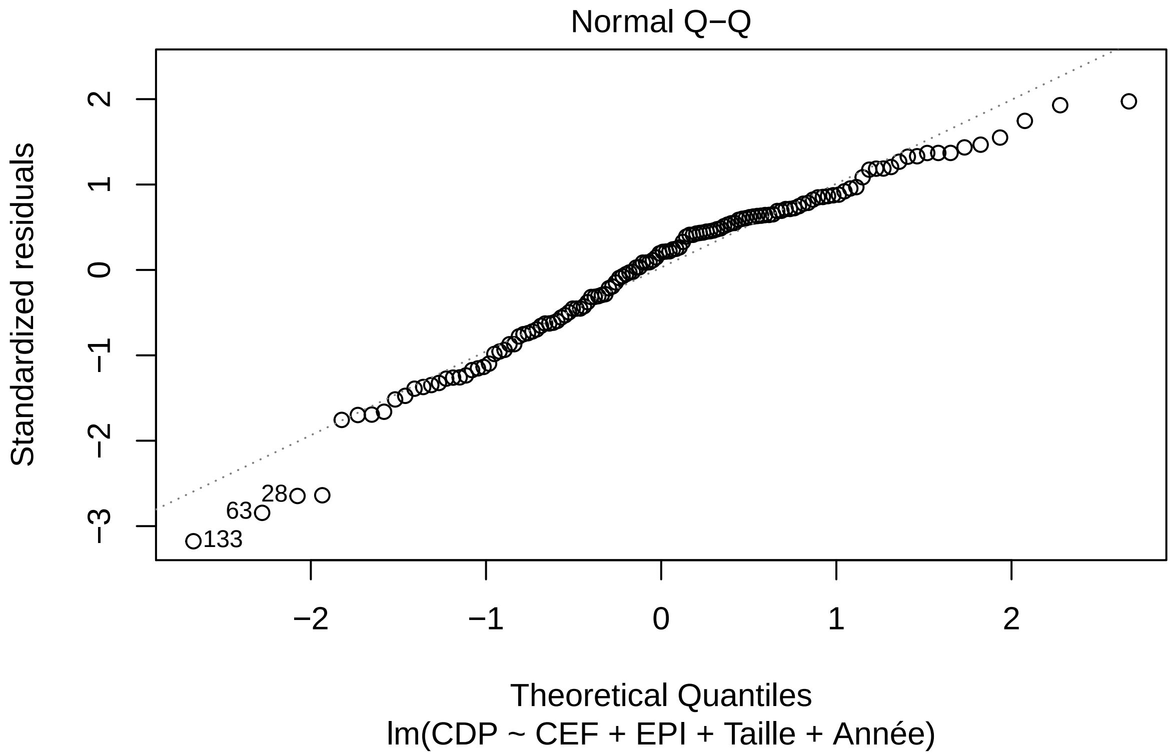
```
## NULL
```

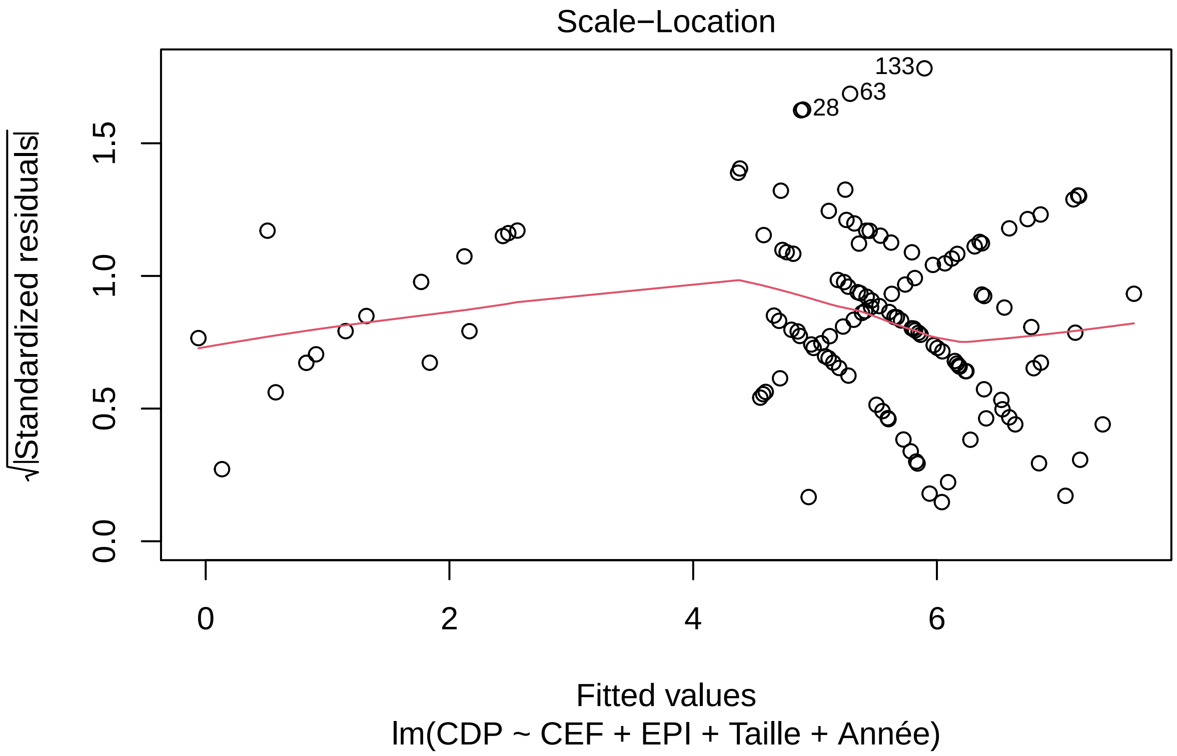
# 6 Approfondissement de la régressions sur CDP

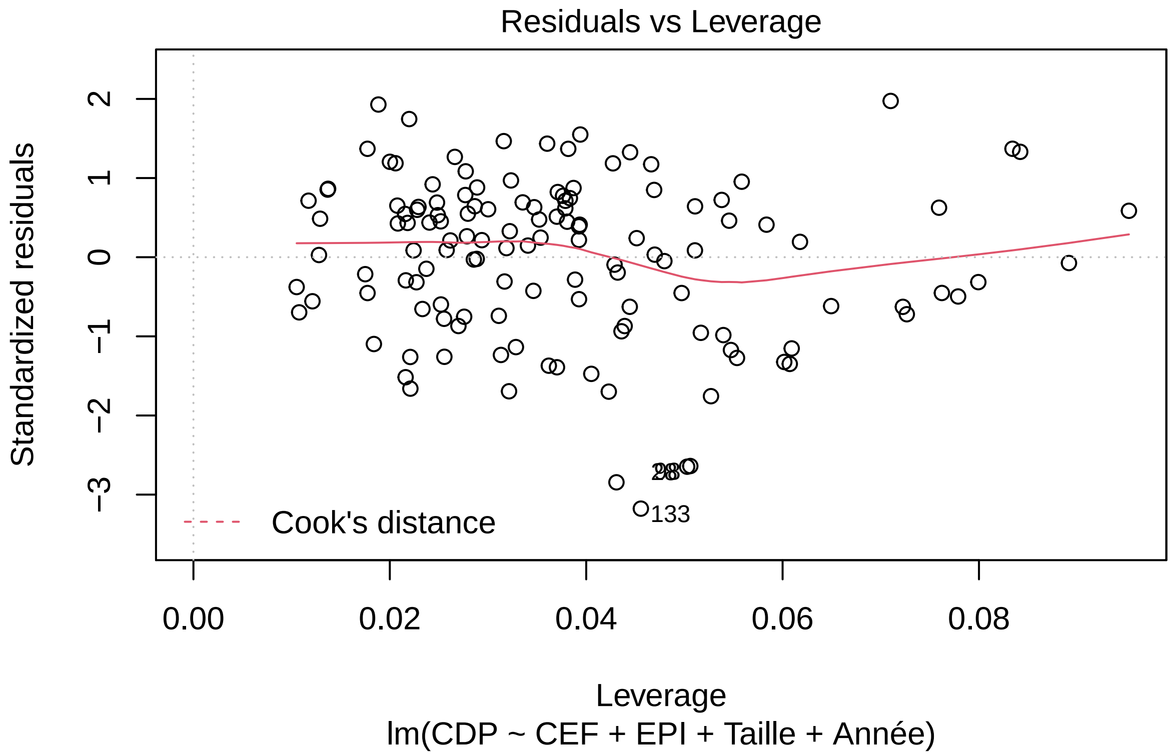
```
reg.CDP <- lm(CDP~CEF+EPI+Taille+Année, data=X)
reg.CDP.1 <- lm(CDP~FEF+CEF+EPI+Taille+Année, data=X)
reg.CDP.2 <- lm(CDP~I(-1.26*FEF+2.66*CEF)+EPI+Taille+Année, data=X)

plot(reg.CDP)
```


Residuals vs Fitted
Residuals
4
2
0
−2
−4
−6
0
2
4
6
28
63
133
Fitted values
lm(CDP ~ CEF + EPI + Taille + Année)


Normal Q−Q
Standardized residuals
2
1
0
−1
−2
−3
28
63
133
−2
−1
0
1
2
Theoretical Quantiles
lm(CDP ~ CEF + EPI + Taille + Année)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
0
2
4
6
Fitted values
lm(CDP ~ CEF + EPI + Taille + Année)
133
63
28



```
X[c(90,18,63),1:2] # Quelques outliers
```

```
##                                                    X1 Année
## 90 SMBC Group (Sumitomo Mitsui Banking Corporation)    18
## 18  ICBC (Industrial and Commercial Bank of China)     16
## 63                                     BPCE/Natixis    17
```

```
X[63,]
```

```
##                X1 Année 2016 2017 2018 2019      FEF      CEF CDP GRI IV ERSE
## 63 BPCE/Natixis    17    0    1    0    0 9.811167 8.967667   0   2  5    0
##    ERSEc ECL ECLc COT EUR  EPI   Taille Taille.1     Croiss Renta
## 63     0   0    0   0   1 88.2 12.10034        0 0.01999676  0.24
```

```
summary(reg.CDP)
```

```
##
## Call:
## lm(formula = CDP ~ CEF + EPI + Taille + Année, data = X)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.898 -1.181  0.377  1.294  3.631
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.34280   11.26028   0.297 0.767054
## CEF           1.62888    0.43817   3.717 0.000301 ***
## EPI           0.14011    0.02192   6.393 2.84e-09 ***
## Taille       -2.60030    0.83534  -3.113 0.002290 **
## Année         0.37910    0.16787   2.258 0.025632 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.9 on 127 degrees of freedom
##   (8 observations deleted due to missingness)
## Multiple R-squared:  0.4152, Adjusted R-squared:  0.3968
## F-statistic: 22.54 on 4 and 127 DF,  p-value: 4.38e-14
```

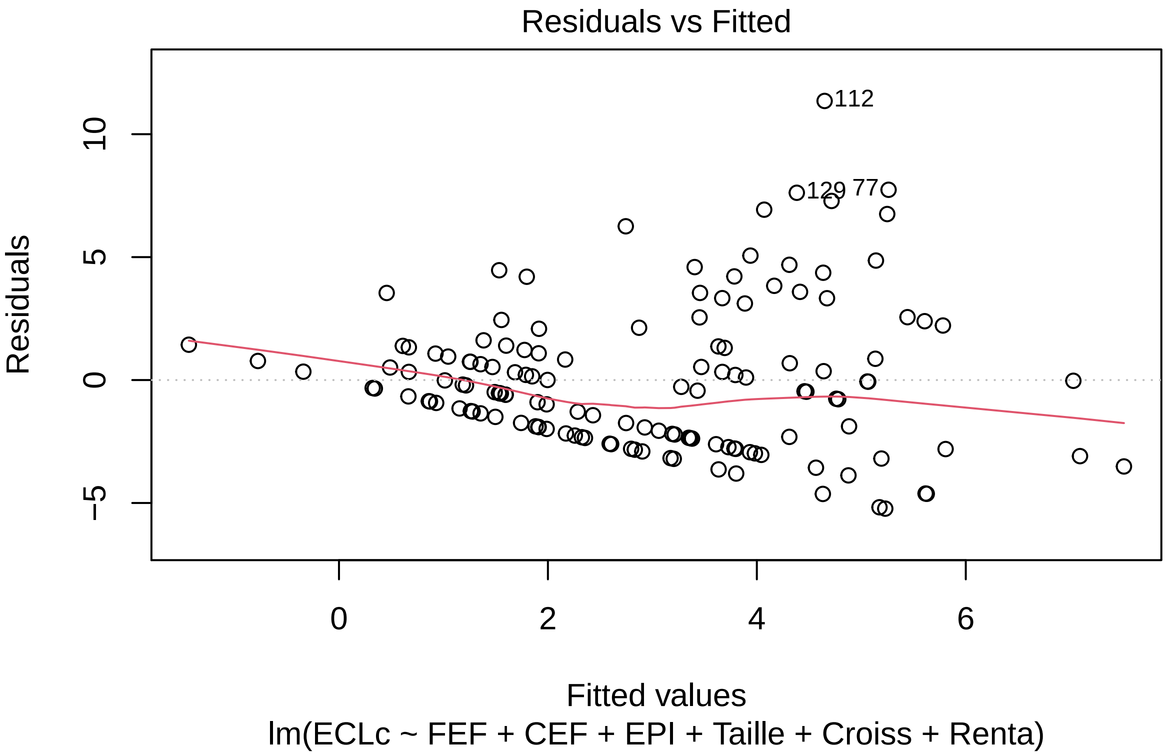
```
summary(reg.CDP.2) # passe de 0.30 à 0.40 en r2.adj
```

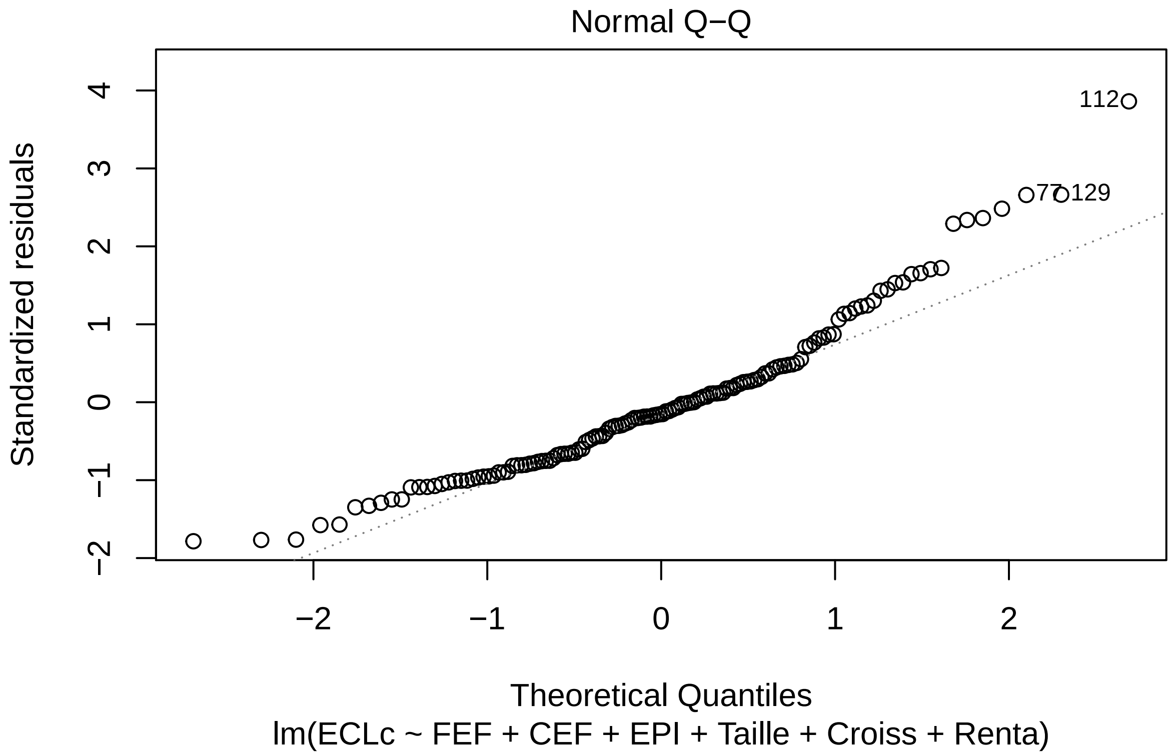
```
##
## Call:
## lm(formula = CDP ~ I(-1.26 * FEF + 2.66 * CEF) + EPI + Taille +
##     Année, data = X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8651 -1.1916  0.3385  1.3022  3.7185
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   5.23288   11.22465   0.466 0.641874
## I(-1.26 * FEF + 2.66 * CEF)  0.99923    0.25774   3.877 0.000169 ***
## EPI                           0.14223    0.02181   6.523 1.49e-09 ***
## Taille                       -2.56342    0.82287  -3.115 0.002273 **
## Année                         0.39740    0.16717   2.377 0.018933 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.892 on 127 degrees of freedom
##   (8 observations deleted due to missingness)
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4019
## F-statistic: 23.01 on 4 and 127 DF,  p-value: 2.573e-14
```

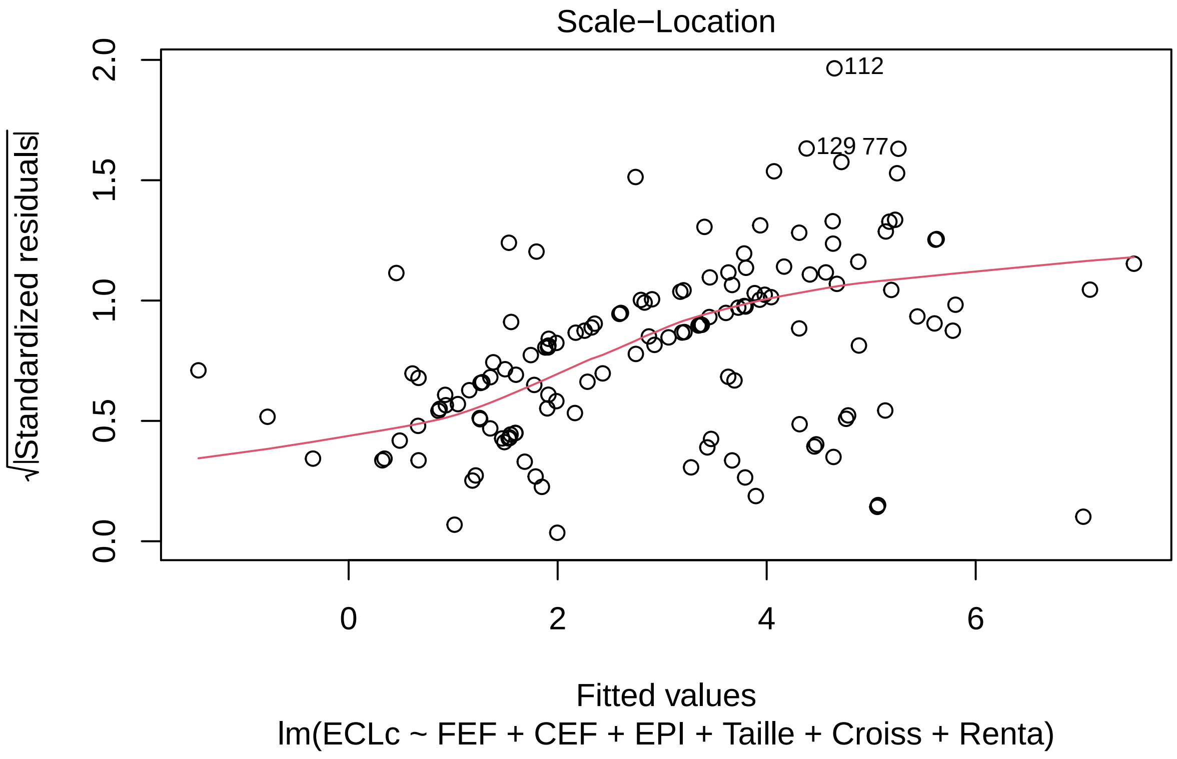
# 7 Approfondissement des autres regressions

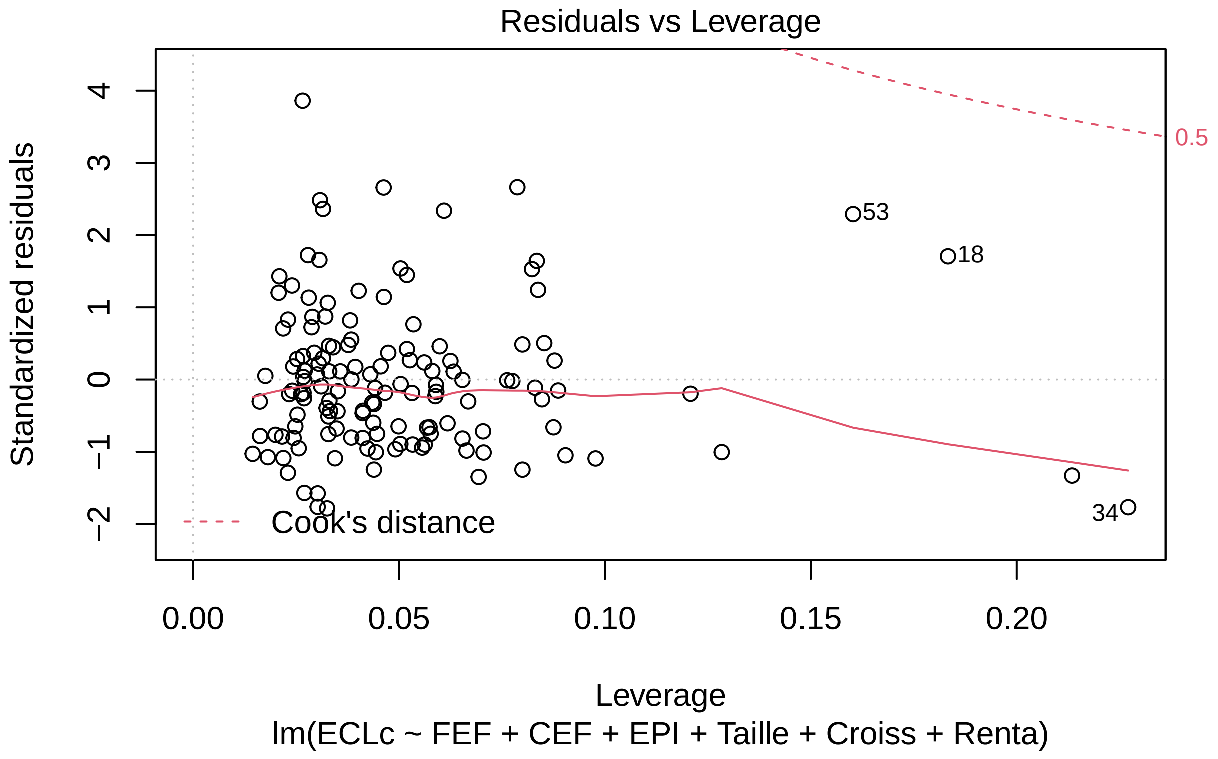
et valeurs extrêmes

```
reg.ECLc <- lm(ECLc~FEF+CEF+EPI+Taille+Croiss+Renta, data=X)
plot(reg.ECLc)
```


Residuals vs Fitted
112
129
77
10
5
0
−5
Residuals
0
2
4
6
Fitted values
lm(ECLc ~ FEF + CEF + EPI + Taille + Croiss + Renta)


Normal Q−Q
Standardized residuals
−2
−1
0
1
2
3
4
−2
−1
0
1
2
112
77
129
Theoretical Quantiles
lm(ECLc ~ FEF + CEF + EPI + Taille + Croiss + Renta)

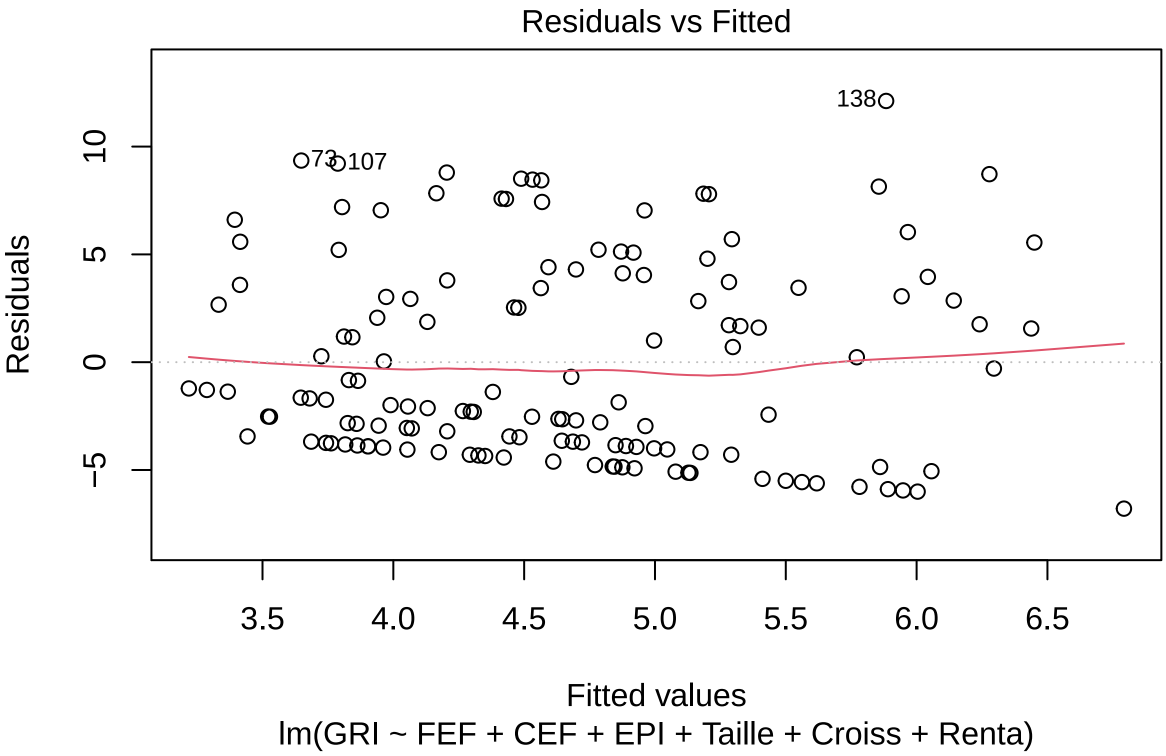

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
2
4
6
112
129
77
Fitted values
lm(ECLc ~ FEF + CEF + EPI + Taille + Croiss + Renta)

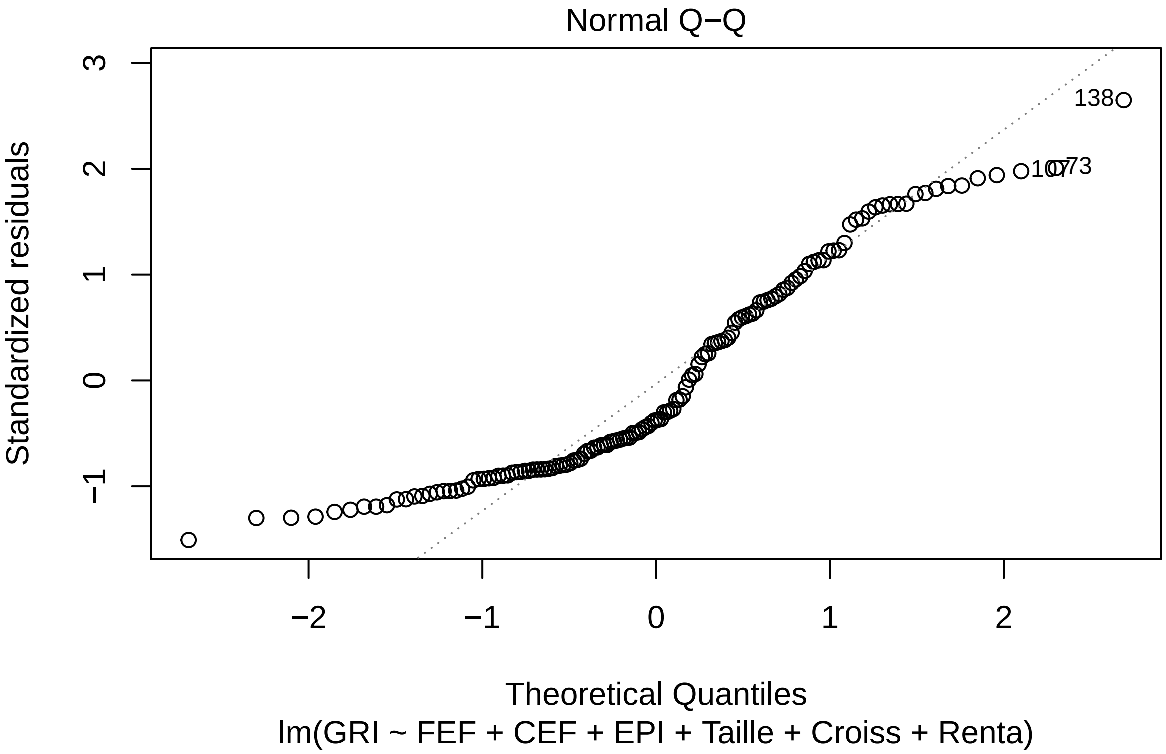


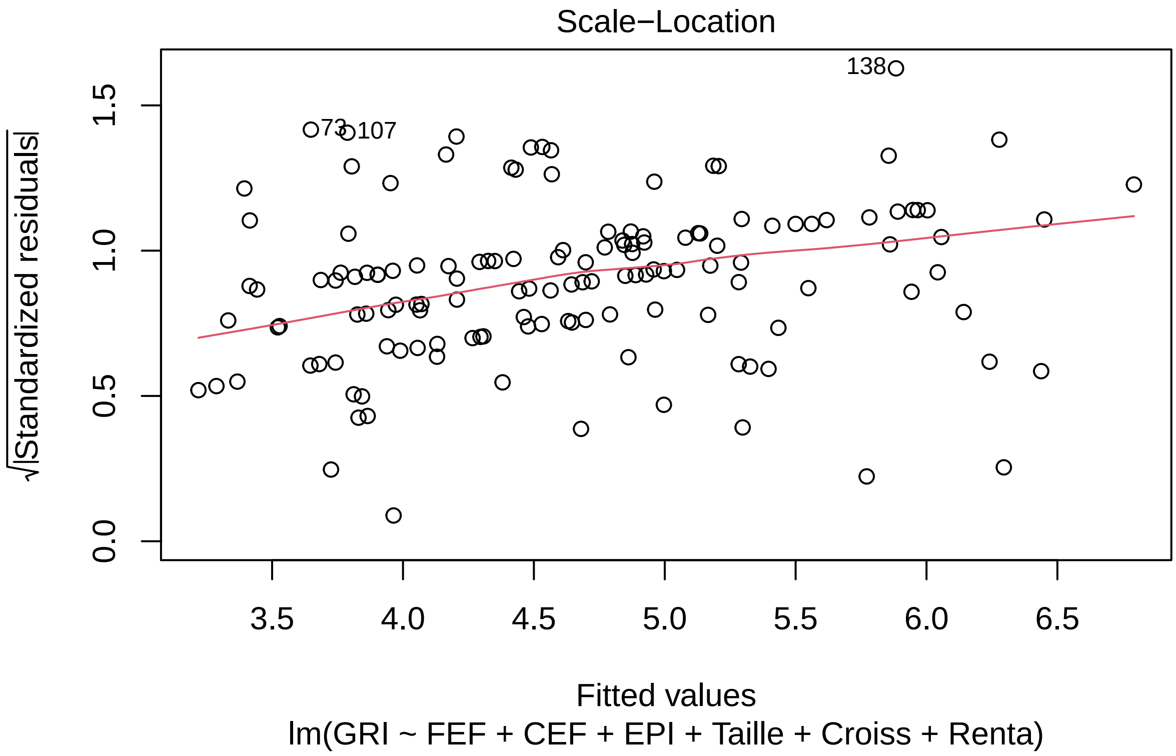
```
X[c(34,18,53),1] # Quelques outliers
```

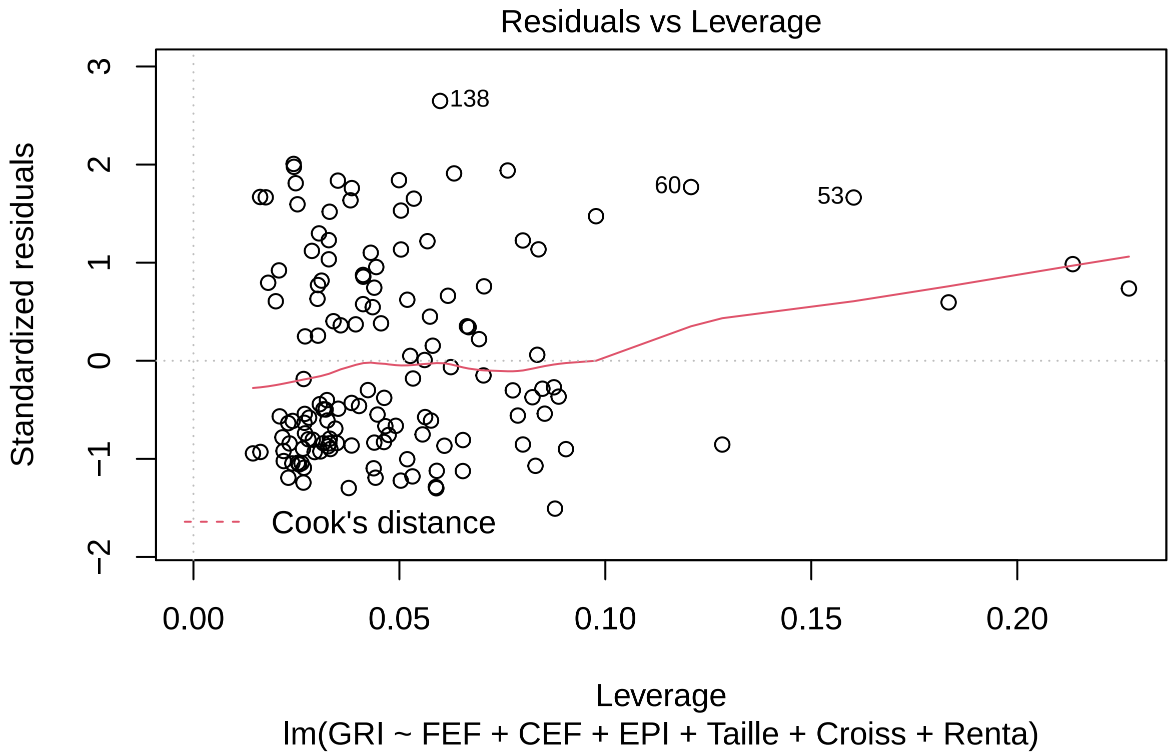
```
## [1] "RBS "
## [2] "ICBC (Industrial and Commercial Bank of China) "
## [3] "ICBC (Industrial and Commercial Bank of China) "
```

```
reg.GRI <- lm(GRI~FEF+CEF+EPI+Taille+Croiss+Renta, data=X)
plot(reg.GRI)
```


Residuals vs Fitted
Residuals
10
5
0
−5
138
73
107
3.5
4.0
4.5
5.0
5.5
6.0
6.5
Fitted values
lm(GRI ~ FEF + CEF + EPI + Taille + Croiss + Renta)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(GRI ~ FEF + CEF + EPI + Taille + Croiss + Renta)
3
2
1
0
−1
−2
−1
0
1
2
138
107
73

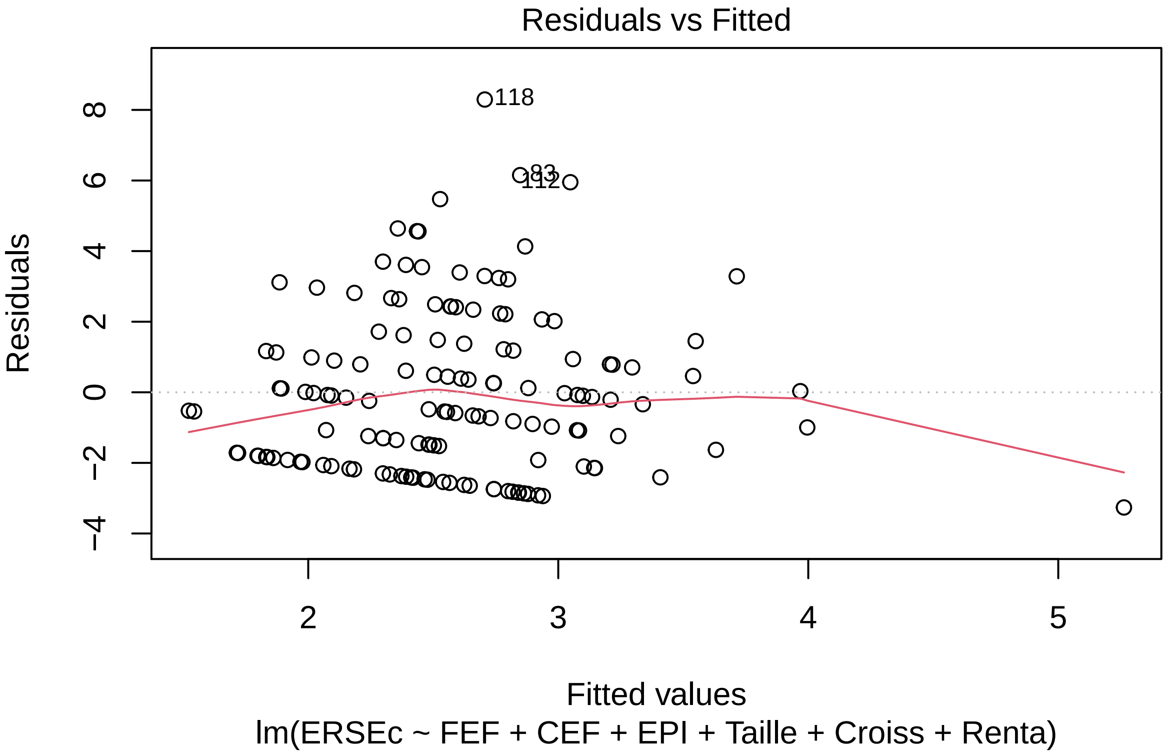

Scale−Location
138
73
107
1.5
1.0
0.5
0.0
√|Standardized residuals|
3.5
4.0
4.5
5.0
5.5
6.0
6.5
Fitted values
lm(GRI ~ FEF + CEF + EPI + Taille + Croiss + Renta)

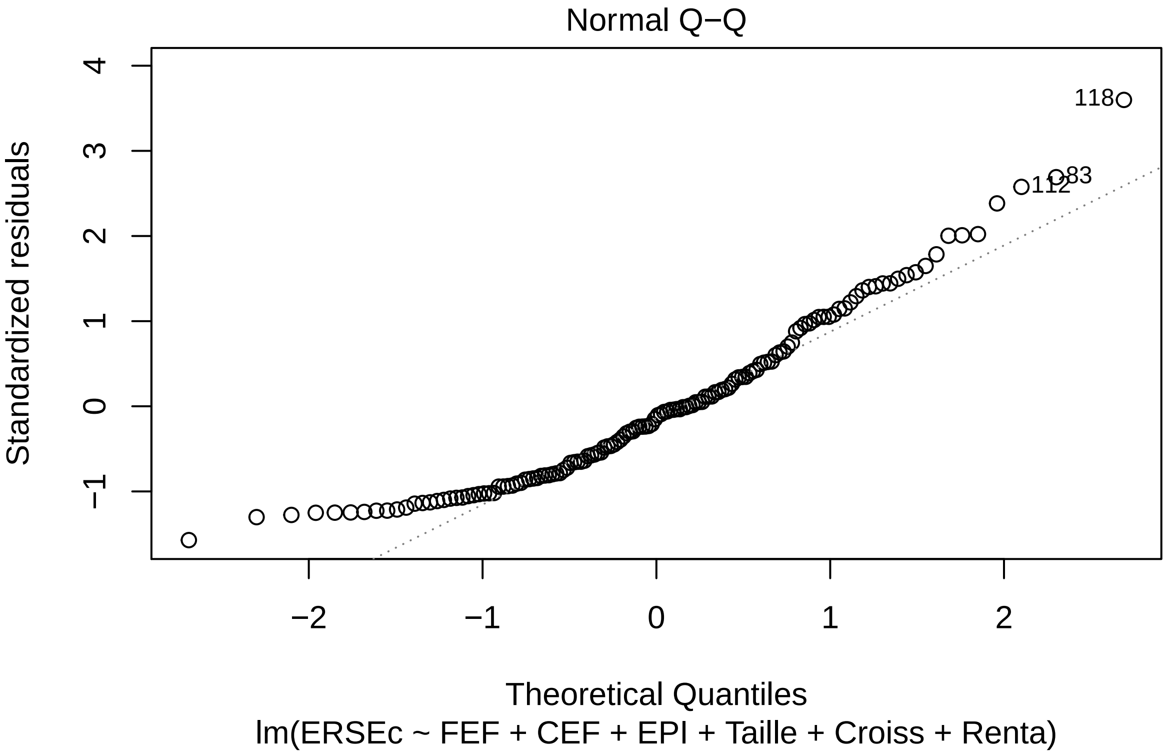


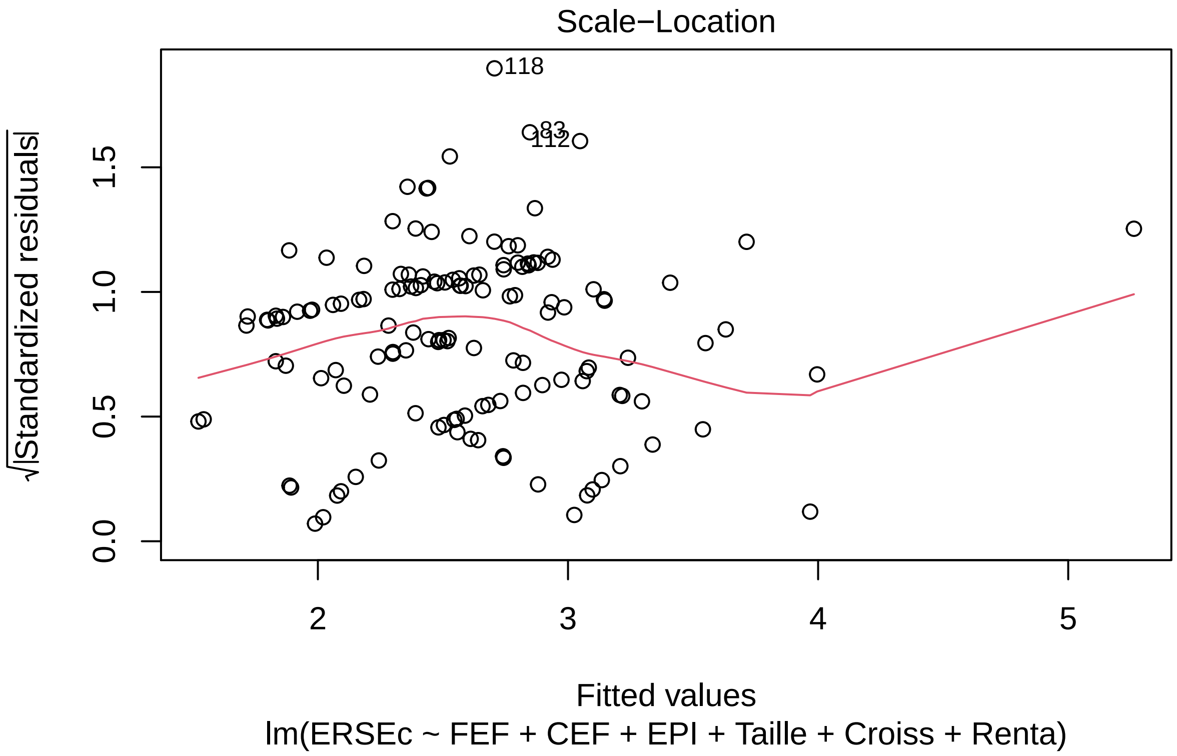
```
X[c(138,60,53),1] # Quelques outliers
```

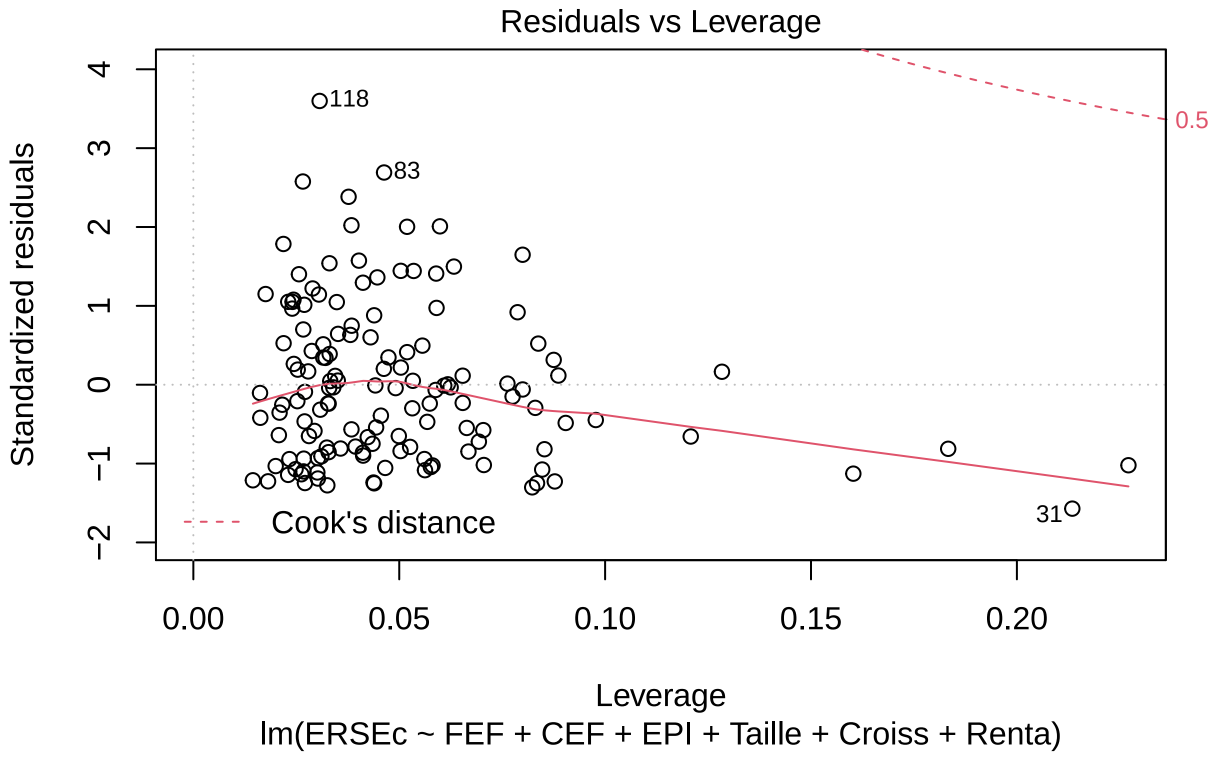
```
## [1] "Intesa Sanpaolo"
## [2] "ING "
## [3] "ICBC (Industrial and Commercial Bank of China) "
```

```
reg.ERSEc <- lm(ERSEc~FEF+CEF+EPI+Taille+Croiss+Renta, data=X)
plot(reg.ERSEc)
```


Residuals vs Fitted
Residuals
Fitted values
lm(ERSEc ~ FEF + CEF + EPI + Taille + Croiss + Renta)
118
83
112
−4
−2
0
2
4
6
8
2
3
4
5


Normal Q−Q
Standardized residuals
4
3
2
1
0
−1
118
112
83
−2
−1
0
1
2
Theoretical Quantiles
lm(ERSEc ~ FEF + CEF + EPI + Taille + Croiss + Renta)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2
3
4
5
118
83
112
Fitted values
lm(ERSEc ~ FEF + CEF + EPI + Taille + Croiss + Renta)



```r
X[c(118,83,31),1] # Quelques outliers
```

```
## [1] "BNP Paribas" "BNP Paribas" "UniCredit "
```

Y : CDP et ECLc / GRI et ERSEc X : FEF ou CEF ; EPI ; Taille ; Croiss ; Renta

# 8 Ridge regression

## 8.1 Réduire les données

Nous devons travailler sur des données centrées réduites pour éviter que l'écrétage soit induit en erreur par des disparités d'échelles.

```r
X.sc <- scale(X[,c('FEF','CEF','EPI','COT','Taille','Croiss','Renta','Année','EUR')])
Y.sc <- scale(X[,c('CDP','ECLc','GRI','ERSEc')])

#names(X)
colnames(X.sc)
```

```
## [1] "FEF"    "CEF"    "EPI"    "COT"    "Taille" "Croiss" "Renta"  "Année"
## [9] "EUR"
```

```r
colnames(Y.sc)
```

```
## [1] "CDP"   "ECLc"  "GRI"   "ERSEc"
```

## 8.2 Load glmnet library

```
#install.packages('glmnet')
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-1
```

## 8.3 Régression Ridge sur toutes les variables Y

```
for(i in 1:ncol(Y.sc)){
cat('-------------------------------------\n')

cat("Ridge regression on variable ", colnames(Y.sc)[i],'\n')
    # detect NA
  which(is.na(Y.sc[,]))  -> NA.ind

# Ajustement d'elastic cv
enet <- cv.glmnet(X.sc[-NA.ind,], Y.sc[-NA.ind,i], alpha=0)

# Fit the Ridge regression
regenet <- glmnet(X.sc[-NA.ind,], Y.sc[-NA.ind,i],
                  alpha=0, # Régression ridge
                  lambda = enet$lambda.1se)
print(coef(regenet))
#plot(regenet)
}
```

```
## -------------------------------------
## Ridge regression on variable  CDP
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) -0.006412076
## FEF          0.057880371
## CEF          0.086915668
## EPI          0.211641609
## COT          0.010293688
## Taille      -0.152369300
## Croiss      -0.043944606
## Renta       -0.059993714
## Année        0.016250845
## EUR          0.041309656
## -------------------------------------
## Ridge regression on variable  ECLc
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept)  0.01710904
## FEF          0.11533062
## CEF          0.13348839
## EPI         -0.07848152
## COT         -0.04714367
## Taille       0.23500553
## Croiss      -0.03478919
## Renta       -0.12812448
## Année        0.10638887
## EUR          0.27784517
## -------------------------------------
## Ridge regression on variable  GRI
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept) -0.0191083967
## FEF         -0.0009490491
## CEF         -0.0004649658
## EPI          0.0005381925
## COT         -0.0001400982
## Taille      -0.0009642353
## Croiss      -0.0004273118
## Renta       -0.0002733365
## Année       -0.0004445308
## EUR          0.0006557906
## -------------------------------------
## Ridge regression on variable  ERSEc
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) -0.017669107
## FEF          0.023139405
## CEF          0.043086676
## EPI         -0.000240310
## COT         -0.014062435
## Taille       0.025114013
## Croiss       0.002334328
## Renta       -0.066883529
## Année        0.358650046
## EUR          0.231402512
```

## 8.4 Détail : Elasticnet on CDP

### 8.4.1 Déterminer le $\lambda$

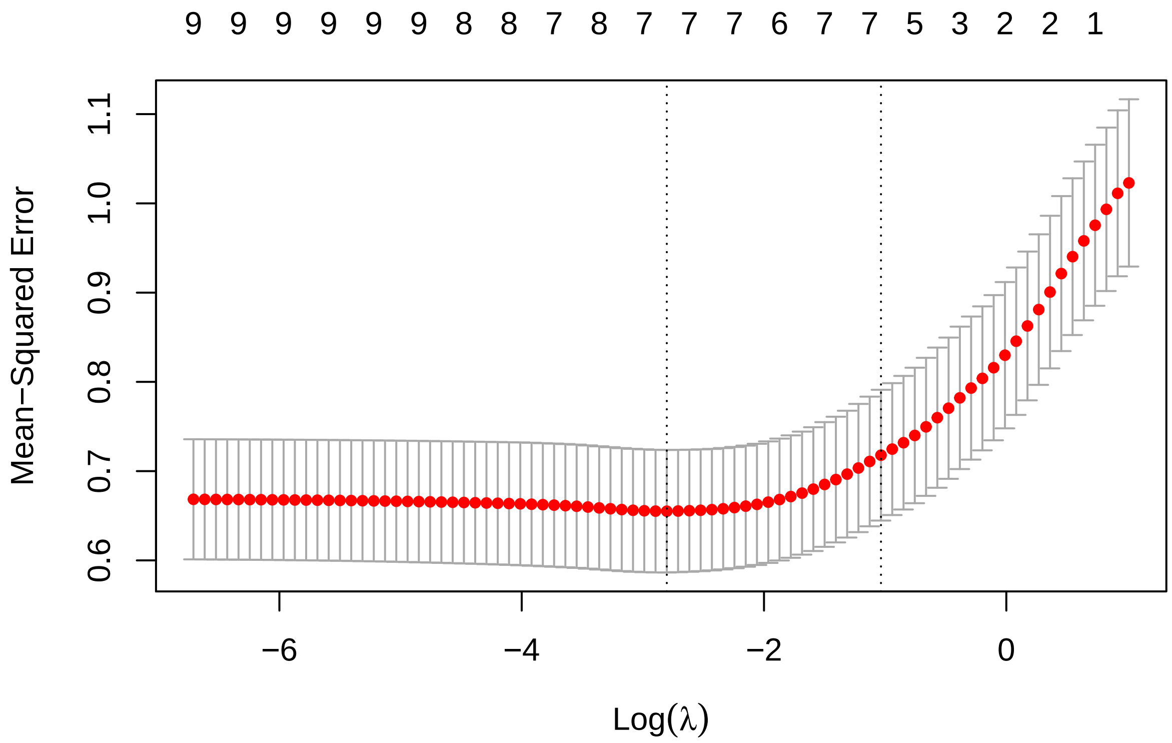
```r
# Found NA
which(is.na(Y.sc[,]), arr.ind = TRUE)[,1] -> NA.CDP
NA.CDP
```

```
##  23  25  54  60  71  95 106 130
##  23  25  54  60  71  95 106 130
```

```r
# Fit elastic cv
enet <- cv.glmnet(X.sc[-NA.CDP,], Y.sc[-NA.CDP,1], alpha=0.2) # manque des CDP à voir
enet
```

```
##
## Call:  cv.glmnet(x = X.sc[-NA.CDP, ], y = Y.sc[-NA.CDP, 1], alpha = 0.2)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure      SE Nonzero
## min 0.0607    42  0.6551 0.06865       7
## 1se 0.3554    23  0.7179 0.07326       7
```

```
plot(enet)
```



La validation croisée nous propose un $\lambda$ optimum à 0.355379.

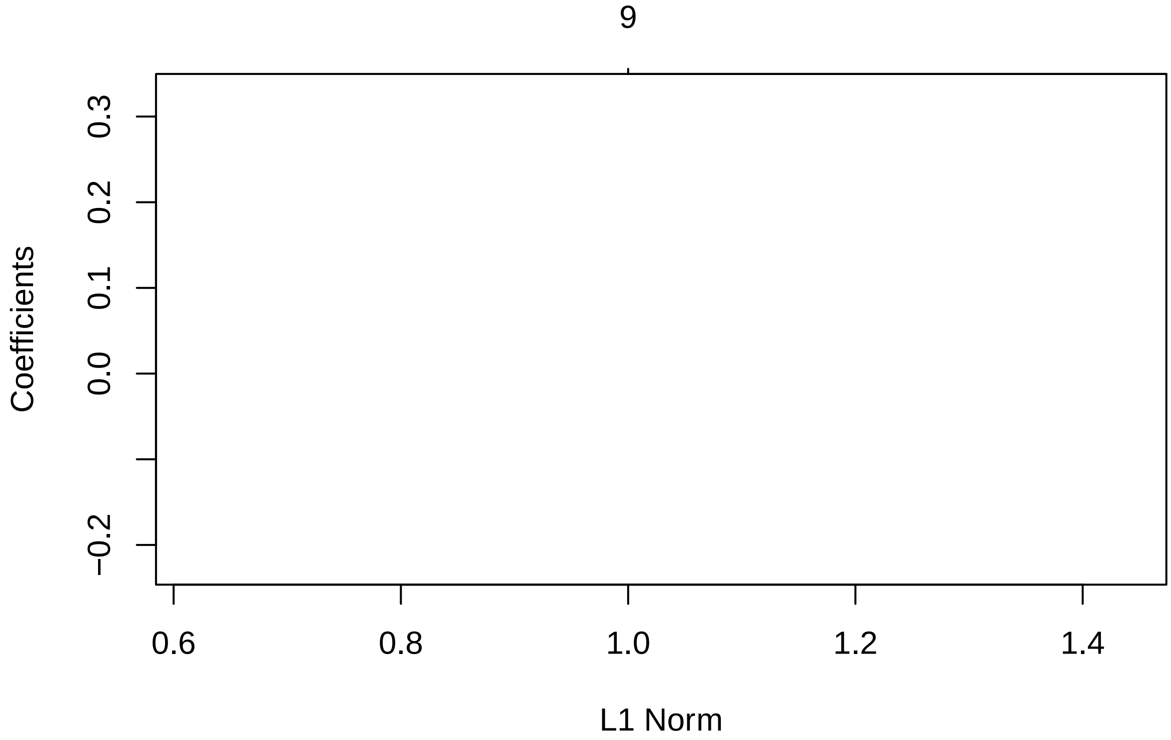
### 8.4.2 Ajustement CDP

On procèdera à l'ajustement avec ce `lambda`

```
regenet <- glmnet(X.sc[-NA.CDP,], Y.sc[-NA.CDP,1], alpha=0,
                  lambda = enet$lambda.1se)
coef(regenet)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) -0.009851716
## FEF          0.071864809
## CEF          0.163346383
## EPI          0.327673331
## COT          0.014633740
## Taille      -0.224272664
## Croiss      -0.064535798
## Renta       -0.051607058
## Année        0.067264173
## EUR          0.043861920
```

```
plot(regenet)
```



On observe un maintien de FEF et de CEF qui sont fortement colinéaires. Aussi l'introduction de CEF dans un modèle sortira automatiquement FEF, et inversement. Il est envisageable de les considérer comme une même variable, ou bien d'exclure FEF selon la méthode habituelle.

Les variables COT, et les 4 dernières sont peu pertinentes.

# 9 Bootstrap

## 9.1 Bootstrap sur les modèles issus de la sélection LEAPS

Définition des modèles sélectionnés :

```
# liste des formules
formules <- c(paste('CDP~CEF+EPI+Taille+Croiss+Année'),
       paste('ECLc~FEF:CEF+EPI+COT+Taille+EUR'),
       paste('ERSEc~CEF+EUR+Année')
)
```

### 9.1.1 CDP

Fragilité de l'année. Le reste est confirmé.

```
set.seed(1)
f <- formula(paste('CDP~CEF+EPI+Taille+Croiss+Année'))
bootstrap(f, data=X)
```

```
## -----------------------------------------
## Regression : CDP ~ CEF + EPI + Taille + Croiss + Année
##             Estim   pval stars Estim_2.5 Estim_97.5 pval_90 stars_90
## (Intercept)  2.22 0.8433          -18.23      24.12  0.9218
## CEF          1.72 0.0001   ***      0.64       2.86  0.0161        *
## EPI          0.14 0.0000   ***      0.09       0.18  0.0000      ***
## Taille      -2.60 0.0022    **     -4.28      -1.06  0.0494        *
## Croiss      -3.69 0.1282           -8.27       0.69  0.6193
## Année        0.41 0.0162     *      0.03       0.73  0.4014
##  r2 ajusté :  0.4031193   quantile 5% = 0.3004187     quantile 50% = 0.4222798
```

### 9.1.2 ECLc

Fragilité de la cotation et de la taille. Le reste est confirmé.

```r
set.seed(1)
f <- formula(paste('ECLc~FEF:CEF+EPI+COT+Taille+EUR'))
bootstrap(f, data=X)
```

```
## -----------------------------------------
## Regression : ECLc ~ FEF:CEF + EPI + COT + Taille + EUR
##              Estim   pval stars Estim_2.5 Estim_97.5 pval_90 stars_90
## (Intercept) -42.30 0.0034    **    -69.02     -21.96  0.0400        *
## EPI          -0.12 0.0004   ***     -0.18      -0.05  0.0308        *
## COT          -0.48 0.0434     *     -0.90      -0.01  0.4183
## Taille        2.25 0.0653     .      0.26       4.69  0.3752
## EUR           5.83 0.0000   ***      4.06       7.71  0.0000      ***
## FEF:CEF       0.26 0.0000   ***      0.15       0.37  0.0001      ***
##  r2 ajusté :  0.393604    quantile 5% = 0.3121403     quantile 50% = 0.4117321
```

### 9.1.3 ERSEc

Tout est confirmé.

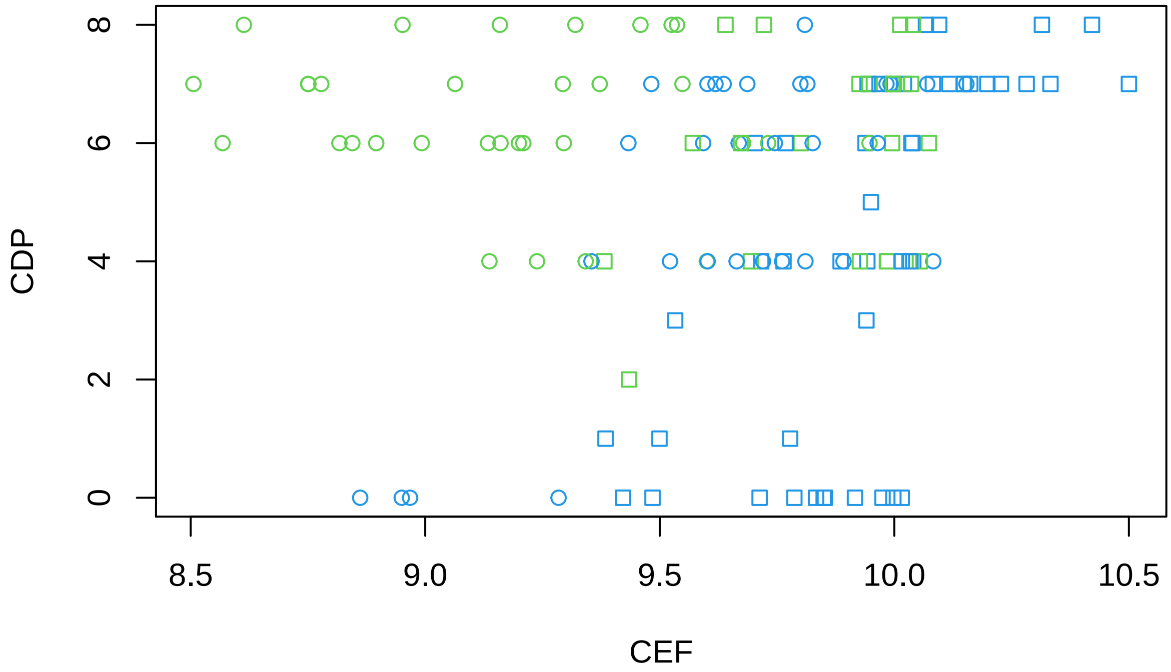
```r
set.seed(1)
f <- formula(paste('ERSEc~CEF+EUR+Année'))
bootstrap(f, data=X)
```

```
## -----------------------------------------
## Regression : ERSEc ~ CEF + EUR + Année
##              Estim  pval stars Estim_2.5 Estim_97.5 pval_90 stars_90
## (Intercept) -35.35 0e+00   ***    -44.52     -25.72  0.0000      ***
## CEF           1.54 2e-04   ***      0.74       2.35  0.0165        *
## EUR           2.54 0e+00   ***      1.87       3.21  0.0000      ***
## Année         1.24 0e+00   ***      1.02       1.47  0.0000      ***
##  r2 ajusté :  0.5371633   quantile 5% = 0.4563583     quantile 50% = 0.5483944
```

# 10 EUR vs NON EUR :

Comparons les régressions entre les banques EUR et non EUR. Le graphique suivant représente le CDP en fonction du CEF. Les couleurs représentent la taille, et la forme des points l'appartenance à l'europe ou pas : le champs d'investigation est caractérisé par les critères de tailles et d'appartenance à l'europe. Dans certains sous-groupes, il est possible que la variable CEF perde de son pouvoir explicatif.

```r
plot(CDP~ CEF, data=X, col=Taille, pch=EUR)
```



## 10.1 Avec les formules précédentes

```
for(frm in formules){
  f <- formula(frm)
  cat('-----------------------------------------\n')
  cat('Regression : ');print(f)
  reg.eur <- lm(formula = f, data=X.eur)
  reg.noneur <- lm(formula = f, data=X.noneur)

    print(cbind(Est.eur = round(summary(reg.eur)$coefficients[,c(1)],3),
                pval.eu = round(summary(reg.eur)$coefficients[,c(4)],3),
                star.eu = star( summary(reg.eur)$coefficients[,c(4)]),
                Est.noneur  = round(summary(reg.noneur)$coefficients[,c(1)],3),
                pval.noneur = round(summary(reg.noneur)$coefficients[,c(4)],3),
                star.noneur = star( summary(reg.noneur)$coefficients[,c(4)])
                ), quote=FALSE
    )

  cat('adj.r.squared = ',broom::glance(reg.eur)[[2]],'     p.value = ',broom::glance(reg.eur)[['p.value
      'adj.r.squared = ',broom::glance(reg.noneur)[[2]],'     p.value = ',broom::glance(reg.noneur)[['p
}
```

```
## -----------------------------------------
## Regression : CDP ~ CEF + EPI + Taille + Croiss + Année
##             Est.eur pval.eu star.eu Est.noneur pval.noneur star.noneur
## (Intercept) 44.962  0.027   *       -42.66     0.004       **
## CEF         1.716   0.051   .       2.507      0.018       *
## EPI         -0.114  0.186           0.201      0           ***
## Taille      -2.968  0.104           -0.837     0.409
## Croiss      -1.852  0.634           -1.651     0.584
## Année       -0.568  0.069   .       1.027      0           ***
## adj.r.squared =  0.05403872      p.value =  0.1474141     adj.r.squared =  0.6640603      p.value =
## -----------------------------------------
## Regression : ECLc ~ FEF:CEF + EPI + COT + Taille + EUR
##             Est.eur pval.eu star.eu Est.noneur pval.noneur star.noneur
## (Intercept) -41.382 0.149           -32.485    0.027       *
## EPI         -0.27   0.004   **      -0.076     0.02        *
## COT         -0.837  0.007   **      0.531      0.158
## Taille      3.323   0.221           2.531      0.031       *
## FEF:CEF     0.315   0       ***     0.08       0.188
## adj.r.squared =  0.4516287      p.value =  1.411825e-08   adj.r.squared =  0.2887601      p.value =
## -----------------------------------------
## Regression : ERSEc ~ CEF + EUR + Année
##             Est.eur pval.eu star.eu Est.noneur pval.noneur star.noneur
## (Intercept) -36.801 0       ***     -34.125    0           ***
## CEF         1.39    0.009   **      1.944      0.004       **
## Année       1.555   0       ***     0.947      0           ***
## adj.r.squared =  0.5203301      p.value =  1.595445e-11   adj.r.squared =  0.4264455      p.value =
```

## 10.2 LEAPS eur

Sélection de modèle sur les banques européennes uniquement

```
for(y in c('CDP','ECLc','GRI','ERSEc')){
  f <- formula(paste(y,'~CEF+EPI+COT+Taille+Croiss+Renta+Année'))
 select.lm(f, data=X.eur, nvmax=7)
}
```

```
## --------------------------------------------
## Regression : CDP ~ CEF + EPI + COT + Taille + Croiss + Renta + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 11.3342  0.0047   **
## Année       -0.3183  0.153
## adj.r.squared =  0.01733506      p.value =  0.1530238
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 45.5208  0.0314   *
## CEF         1.7423   0.0607   .
## EPI         -0.1071  0.2432
## COT         0.1536   0.4649
## Taille      -3.0691  0.0978   .
## Croiss      -3.4365  0.4005
## Renta       0.947    0.2976
## Année       -0.6083  0.0717   .
## adj.r.squared =  0.05162319      p.value =  0.1928979
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 11.3342  0.0047   **
## Année       -0.3183  0.153
## adj.r.squared =  0.01733506      p.value =  0.1530238
##
## --------------------------------------------
## Regression : ECLc ~ CEF + EPI + COT + Taille + Croiss + Renta + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -31.4858 0.0031   **
## CEF         6.434    0        ***
## EPI         -0.2786  0.0041   **
## COT         -0.9179  0.0048   **
## adj.r.squared =  0.4194983      p.value =  2.802454e-08
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -73.5423 0.005    **
## CEF         4.5693   9e-04    ***
## EPI         -0.2998  0.002    **
## COT         -0.8627  0.0069   **
## Taille      5.1123   0.0604   .
## Renta       -1.7222  0.1672
## adj.r.squared =  0.442993      p.value =  5.835936e-08
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -69.6892 0.0077   **
## CEF         4.945    3e-04    ***
## EPI         -0.2795  0.0035   **
## COT         -0.8926  0.0055   **
## Taille      4.331    0.105
## adj.r.squared =  0.4345615      p.value =  3.592097e-08
##
## --------------------------------------------
## Regression : GRI ~ CEF + EPI + COT + Taille + Croiss + Renta + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 91.9903  0.0181   *
## Taille      -7.2     0.0253   *
## adj.r.squared =  0.0594776      p.value =  0.02532071
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 91.9903  0.0181   *
## Taille      -7.2     0.0253   *
## adj.r.squared =  0.0594776      p.value =  0.02532071
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 91.9903  0.0181   *
## Taille      -7.2     0.0253   *
## adj.r.squared =  0.0594776      p.value =  0.02532071
##
## --------------------------------------------
## Regression : ERSEc ~ CEF + EPI + COT + Taille + Croiss + Renta + Année
## Selected by  bic      --------------------
##             Estimate Pr(>|t|) star
## (Intercept) -15.9016 0.0791   .
## CEF         1.8672   6e-04    ***
## EPI         -0.2019  0.0041   **
## Année       1.0706   0        ***
## adj.r.squared =  0.5721088      p.value =  1.847026e-12
##
## Selected by  adjr2    --------------------
##             Estimate Pr(>|t|) star
## (Intercept) -27.915  0.0802   .
## CEF         1.4943   0.0445   *
## EPI         -0.2282  0.0018   **
## COT         -0.2285  0.1968
## Taille      1.599    0.2772
## Année       1.0068   1e-04    ***
## adj.r.squared =  0.5790275      p.value =  1.374688e-11
##
## Selected by  cp   --------------------
##             Estimate Pr(>|t|) star
## (Intercept) -15.9016 0.0791   .
## CEF         1.8672   6e-04    ***
## EPI         -0.2019  0.0041   **
## Année       1.0706   0        ***
## adj.r.squared =  0.5721088      p.value =  1.847026e-12
```

### 10.3 LEAPS non eur

```
for(y in c('CDP','ECLc','GRI','ERSEc')){
  f <- formula(paste(y,'~CEF+EPI+COT+Taille+Croiss+Renta+Année'))
 select.lm(f, data=X.noneur, nvmax=7)
}
```

```
## --------------------------------------------
## Regression : CDP ~ CEF + EPI + COT + Taille + Croiss + Renta + Année
## Selected by  bic      --------------------
##             Estimate Pr(>|t|) star
## (Intercept) -54.7519 0        ***
## CEF         2.7283   0.0021   **
## EPI         0.2139   0        ***
## COT         -0.8172  0.0044   **
## Année       1.0441   0        ***
## adj.r.squared =  0.7045034      p.value =  3.955286e-17
##
## Selected by  adjr2    --------------------
##             Estimate Pr(>|t|) star
## (Intercept) -42.3391 0.0022   **
## CEF         3.248    0.0012   **
## EPI         0.1871   0        ***
## COT         -0.8483  0.0032   **
## Taille      -1.1061  0.2467
## Année       0.9264   1e-04    ***
## adj.r.squared =  0.7061903      p.value =  1.40901e-16
##
```

```
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -54.7519 0        ***
## CEF         2.7283   0.0021   **
## EPI         0.2139   0        ***
## COT         -0.8172  0.0044   **
## Année       1.0441   0        ***
## adj.r.squared =  0.7045034      p.value =  3.955286e-17
##
## --------------------------------------------
## Regression : ECLc ~ CEF + EPI + COT + Taille + Croiss + Renta + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -50.7884 0        ***
## Taille      4.3393   0        ***
## adj.r.squared =  0.2302393      p.value =  1.192058e-05
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -38.4901 0.0141   *
## CEF         2.0433   0.0828   .
## EPI         -0.0961  0.0083   **
## COT         0.505    0.1756
## Taille      2.2272   0.0592   .
## Renta       -0.9627  0.2872
## adj.r.squared =  0.2979264      p.value =  2.564575e-05
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -42.2099 0.0071   **
## CEF         2.1855   0.0539   .
## EPI         -0.0822  0.0111   *
## Taille      2.3511   0.0455   *
## adj.r.squared =  0.2865195      p.value =  9.195524e-06
##
## --------------------------------------------
## Regression : GRI ~ CEF + EPI + COT + Taille + Croiss + Renta + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -26.0419 0.1755
## CEF         3.0388   0.117
## adj.r.squared =  0.02095156      p.value =  0.116961
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -26.0419 0.1755
## CEF         3.0388   0.117
## adj.r.squared =  0.02095156      p.value =  0.116961
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -26.0419 0.1755
## CEF         3.0388   0.117
## adj.r.squared =  0.02095156      p.value =  0.116961
```

```
##
## --------------------------------------------
## Regression : ERSEc ~ CEF + EPI + COT + Taille + Croiss + Renta + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -34.1246 0        ***
## CEF         1.9438   0.0041   **
## Année       0.9471   0        ***
## adj.r.squared =  0.4264455      p.value =  1.748665e-09
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -34.1246 0        ***
## CEF         1.9438   0.0041   **
## Année       0.9471   0        ***
## adj.r.squared =  0.4264455      p.value =  1.748665e-09
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -34.1246 0        ***
## CEF         1.9438   0.0041   **
## Année       0.9471   0        ***
## adj.r.squared =  0.4264455      p.value =  1.748665e-09
```

### 10.4 Formules de LEAPS

```
formules.eur <- c(
  ECLc ~ CEF+EPI + COT + Taille,
  ERSEc~ CEF + EPI +  Année
)

formules.noneur <- c(
  CDP ~ CEF + EPI + COT +Année,
  ECLc ~CEF+EPI+ Taille,
  ERSEc ~ CEF +  Année
)
```

## 11 Petites vs Grandes régressions

### 11.1 Avec les formules précédentes

```
for(frm in formules){
  f <- formula(frm)
  cat('-----------------------------------------\n')
  cat('Regression : ');print(f)
  reg.grandes <- lm(formula = f, data=X.grandes)
  reg.petites <- lm(formula = f, data=X.petites)

    print(cbind(Est.grandes = round(summary(reg.grandes)$coefficients[,c(1)],3),
                pval.grandes = round(summary(reg.grandes)$coefficients[,c(4)],3),
                star.grandes = star(summary(reg.grandes)$coefficients[,c(4)]),
                Est.petites  = round(summary(reg.petites)$coefficients[,c(1)],3),
                pval.petites = round(summary(reg.petites)$coefficients[,c(4)],3),
                star.petites = star(summary(reg.petites)$coefficients[,c(4)])
                ), quote=FALSE
    )

  cat('adj.r.squared = ',broom::glance(reg.grandes)[[2]],'     p.value = ',broom::glance(reg.grandes)[[
      'adj.r.squared = ',broom::glance(reg.petites)[[2]],'     p.value = ',broom::glance(reg.petites)[[
}
```

```
## -----------------------------------------
## Regression : CDP ~ CEF + EPI + Taille + Croiss + Année
##             Est.grandes pval.grandes star.grandes Est.petites pval.petites
## (Intercept) -18.227     0.592                     45.83       0.081
## CEF         3.026       0            ***          0.88        0.149
## EPI         0.167       0            ***          -0.063      0.305
## Taille      -2.218      0.39                      -3.442      0.133
## Croiss      0.999       0.736                     -9.707      0.016
## Année       0.442       0.026        *            -0.123      0.694
##             star.petites
## (Intercept) .
## CEF
## EPI
## Taille
## Croiss      *
## Année
## adj.r.squared =  0.6784129      p.value =  1.186073e-14   adj.r.squared =  0.06290077      p.value =
## -----------------------------------------
## Regression : ECLc ~ FEF:CEF + EPI + COT + Taille + EUR
##             Est.grandes pval.grandes star.grandes Est.petites pval.petites
## (Intercept) 67.749      0.229                     -26.255     0.239
## EPI         -0.189      0            ***          -0.122      0.001
## COT         -1.337      0.001        ***          0.355       0.075
## Taille      -6.184      0.157                     2.811       0.162
## EUR         7.255       0            ***          1.999       0.014
## FEF:CEF     0.269       0.001        ***          0.024       0.613
##             star.petites
## (Intercept)
## EPI         ***
## COT         .
## Taille
## EUR         *
## FEF:CEF
## adj.r.squared =  0.4666383      p.value =  8.969891e-09   adj.r.squared =  0.2711434      p.value =
## -----------------------------------------
## Regression : ERSEc ~ CEF + EUR + Année
##             Est.grandes pval.grandes star.grandes Est.petites pval.petites
## (Intercept) -43.851     0            ***          -25.957     0.001
## CEF         2.707       0.001        ***          0.272       0.691
## EUR         2.866       0            ***          1.57        0.016
## Année       1.07        0            ***          1.424       0
##             star.petites
## (Intercept) ***
## CEF
## EUR         *
## Année       ***
## adj.r.squared =  0.576268      p.value =  5.820684e-13    adj.r.squared =  0.5296455      p.value =
```

## 11.2 LEAPS grandes

Sélection de modèle sur les banques européennes uniquement

```
for(y in c('CDP','ECLc','GRI','ERSEc')){
  f <- formula(paste(y,'~CEF+EPI+COT+Taille+Croiss+Renta+EUR+Année'))
 select.lm(f, data=X.grandes, nvmax=7)
}
```

```
## --------------------------------------------
## Regression : CDP ~ CEF + EPI + COT + Taille + Croiss + Renta + EUR + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -46.6102 0        ***
## CEF         2.9077   3e-04    ***
## EPI         0.1849   0        ***
## Année       0.4889   0.0098   **
## adj.r.squared =  0.6848886      p.value =  3.505984e-16
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -4.7998  0.8844
## CEF         2.6829   0.0011   **
## EPI         0.1836   0        ***
## COT         0.2663   0.1974
## Taille      -3.3271  0.1832
## Renta       0.9469   0.1604
## Année       0.5167   0.0094   **
## adj.r.squared =  0.6898852      p.value =  1.461173e-14
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -46.6102 0        ***
## CEF         2.9077   3e-04    ***
## EPI         0.1849   0        ***
## Année       0.4889   0.0098   **
## adj.r.squared =  0.6848886      p.value =  3.505984e-16
##
## --------------------------------------------
## Regression : ECLc ~ CEF + EPI + COT + Taille + Croiss + Renta + EUR + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -35.0654 0.0169   *
## CEF         5.5997   6e-04    ***
## EPI         -0.1823  2e-04    ***
## COT         -1.4728  2e-04    ***
## Renta       -2.3562  0.0462   *
## EUR         6.5462   0        ***
## adj.r.squared =  0.4745902      p.value =  5.66133e-09
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -35.0654 0.0169   *
## CEF         5.5997   6e-04    ***
## EPI         -0.1823  2e-04    ***
## COT         -1.4728  2e-04    ***
## Renta       -2.3562  0.0462   *
## EUR         6.5462   0        ***
## adj.r.squared =  0.4745902      p.value =  5.66133e-09
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -35.0654 0.0169   *
## CEF         5.5997   6e-04    ***
## EPI         -0.1823  2e-04    ***
## COT         -1.4728  2e-04    ***
## Renta       -2.3562  0.0462   *
## EUR         6.5462   0        ***
## adj.r.squared =  0.4745902      p.value =  5.66133e-09
##
## --------------------------------------------
## Regression : GRI ~ CEF + EPI + COT + Taille + Croiss + Renta + EUR + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 14.1675  0.056    .
## Année       -0.5905  0.1602
## adj.r.squared =  0.01451253      p.value =  0.1602059
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 207.1447 0.0143   *
## CEF         2.9594   0.2137
## EPI         -0.1681  0.0626   .
## Taille      -16.2383 0.013    *
## Croiss      14.6316  0.0528   .
## EUR         0.0169   0.9912
## Année       -1.1312  0.0326   *
## adj.r.squared =  0.07205505      p.value =  0.09588089
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 14.1675  0.056    .
## Année       -0.5905  0.1602
## adj.r.squared =  0.01451253      p.value =  0.1602059
##
## --------------------------------------------
## Regression : ERSEc ~ CEF + EPI + COT + Taille + Croiss + Renta + EUR + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -43.8508 0        ***
## CEF         2.7073   7e-04    ***
## EUR         2.8662   0        ***
## Année       1.0696   0        ***
## adj.r.squared =  0.576268      p.value =  5.820684e-13
##
## Selected by  adjr2    -------------------
##             Estimate  Pr(>|t|) star
## (Intercept) -101.9569 0.002    **
## CEF         2.5802    0.0053   **
## EPI         0.0378    0.2605
## COT         -0.251    0.2226
## Taille      4.5066    0.0626   .
## EUR         2.8976    0        ***
## Année       1.1478    0        ***
## adj.r.squared =  0.5850968      p.value =  1.146613e-11
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -43.8508 0        ***
## CEF         2.7073   7e-04    ***
## EUR         2.8662   0        ***
## Année       1.0696   0        ***
## adj.r.squared =  0.576268      p.value =  5.820684e-13
```

## 11.3 LEAPS petites

```
for(y in c('CDP','ECLc','GRI','ERSEc')){
  f <- formula(paste(y,'~CEF+EPI+COT+Taille+Croiss+Renta+EUR+Année'))
 select.lm(f, data=X.petites, nvmax=7)
}
```

```
## --------------------------------------------
## Regression : CDP ~ CEF + EPI + COT + Taille + Croiss + Renta + EUR + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 5.8771   0        ***
## Croiss      -5.7367  0.1078
## adj.r.squared =  0.02489575      p.value =  0.1078425
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 44.8105  0.0838   .
## CEF         0.9438   0.106
## EPI         -0.0466  0.299
## Taille      -3.6977  0.0902   .
## Croiss      -9.9116  0.0129   *
## adj.r.squared =  0.07586612      p.value =  0.06558045
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 40.7369  0.1112
## CEF         1.037    0.0731   .
## Taille      -3.7503  0.0859   .
## Croiss      -8.6961  0.0218   *
## adj.r.squared =  0.07441421      p.value =  0.05066433
##
## --------------------------------------------
## Regression : ECLc ~ CEF + EPI + COT + Taille + Croiss + Renta + EUR + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -10.8843 4e-04    ***
## Renta       -1.4453  0.0033   **
## Année       0.7414   0        ***
## adj.r.squared =  0.2425674      p.value =  3.3891e-05
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -38.0473 0.0603   .
## EPI         -0.0617  0.2059
## COT         0.435    0.0175   *
## Taille      2.9303   0.0745   .
## Croiss      -5.296   0.0865   .
## Renta       -0.7076  0.2957
## EUR         0.643    0.2723
## Année       0.5028   0.0396   *
## adj.r.squared =  0.3290658      p.value =  3.179603e-05
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -47.0448 0.0153   *
## COT         0.4536   0.0131   *
## Taille      2.9482   0.0706   .
## Croiss      -5.694   0.061    .
## Renta       -0.9878  0.0588   .
## Année       0.7458   0        ***
## adj.r.squared =  0.3284816      p.value =  9.470567e-06
##
## --------------------------------------------
## Regression : GRI ~ CEF + EPI + COT + Taille + Croiss + Renta + EUR + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 4        1e-04    ***
## EUR         2.4286   0.0573   .
## adj.r.squared =  0.03817324      p.value =  0.0573365
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -30.9774 0.147
## CEF         3.5351   0.1019
## EUR         5.0234   0.0143   *
## adj.r.squared =  0.06231231      p.value =  0.04325253
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -30.9774 0.147
## CEF         3.5351   0.1019
## EUR         5.0234   0.0143   *
## adj.r.squared =  0.06231231      p.value =  0.04325253
##
## --------------------------------------------
## Regression : ERSEc ~ CEF + EPI + COT + Taille + Croiss + Renta + EUR + Année
## Selected by  bic      -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -23.3544 0        ***
## EUR         1.3708   9e-04    ***
## Année       1.4284   0        ***
## adj.r.squared =  0.5355498      p.value =  2.597297e-12
##
## Selected by  adjr2    -------------------
##             Estimate Pr(>|t|) star
## (Intercept) 8.4862   0.6883
## CEF         0.9894   0.1864
## EPI         -0.1101  0.0483   *
## Taille      -2.2337  0.2247
## EUR         2.7261   0.0011   **
## Année       1.0602   1e-04    ***
## adj.r.squared =  0.5525502      p.value =  3.999963e-11
##
## Selected by  cp   -------------------
##             Estimate Pr(>|t|) star
## (Intercept) -10.5486 0.1859
## EPI         -0.0918  0.0843   .
## EUR         1.8239   2e-04    ***
## Année       1.108    0        ***
## adj.r.squared =  0.5494837      p.value =  4.308401e-12
```

### 11.4 Formules de LEAPS

```
formules.grandes <- c(
  CDP  ~ CEF+EPI+Année ,
  ECLc ~ CEF+EPI+COT+EUR,
  ERSEc~ CEF+EUR+Année
)

formules.petites <- c(
  CDP ~ CEF+Taille+Croiss ,
  ECLc ~ COT + Taille + Renta+ Croiss +Année ,
  ERSEc ~ EPI+EUR+Année
)
```

## 12 Années 0/1 dans toutes les régressions

Pas d'intervention majeure des années, sauf exception.

```
for(annee in 2016:2019){
  for(frm in formules){
    frm <- paste(frm,'+An', annee, sep='')
    f <- formula(frm)
    cat('----------------------------------------\n')
    cat('Regression : ');print(f)
    reg <- lm(formula = f, data=X.all)
    print(cbind(
            round(summary(reg)$coefficients[,c(1,4)],4),
            star = star(summary(reg)$coefficients[,c(4)])
      ), quote=FALSE
    )
    cat('adj.r.squared = ',broom::glance(reg)[[2]],'     p.value = ',broom::glance(reg)[[4]], '\n')
  }
}
```

```
## ----------------------------------------
## Regression : CDP ~ CEF + EPI + Taille + Croiss + Année + An2016
##             Estimate Pr(>|t|) star
## (Intercept) 0.228    0.9849
## CEF         1.6913   2e-04    ***
## EPI         0.1386   0        ***
## Taille      -2.5612  0.0027   **
## Croiss      -3.4013  0.1747
## Année       0.5021   0.0512   .
## An2016      0.3018   0.6316
## adj.r.squared =  0.3994543      p.value =  15.52249
## ----------------------------------------
## Regression : ECLc ~ FEF:CEF + EPI + COT + Taille + EUR + An2016
##             Estimate Pr(>|t|) star
## (Intercept) -43.2012 0.0034   **
## EPI         -0.1193  0.0017   **
## COT         -0.4719  0.0458   *
## Taille      2.327    0.062    .
## EUR         5.7651   0        ***
## An2016      -0.1858  0.7384
## FEF:CEF     0.259    0        ***
## adj.r.squared =  0.3895587      p.value =  15.78402
## ----------------------------------------
## Regression : ERSEc ~ CEF + EUR + Année + An2016
##             Estimate Pr(>|t|) star
## (Intercept) -40.3699 0        ***
## CEF         1.4396   4e-04    ***
## EUR         2.4861   0        ***
## Année       1.5746   0        ***
## An2016      1.0973   0.0267   *
## adj.r.squared =  0.5504566      p.value =  43.55066
## ----------------------------------------
## Regression : CDP ~ CEF + EPI + Taille + Croiss + Année + An2017
##             Estimate Pr(>|t|) star
## (Intercept) 1.7424   0.8775
## CEF         1.6951   2e-04    ***
## EPI         0.1406   0        ***
## Taille      -2.531   0.0031   **
## Croiss      -3.346   0.179
## Année       0.3917   0.0237   *
## An2017      -0.2636  0.5252
## adj.r.squared =  0.4002921      p.value =  15.57328
## ----------------------------------------
## Regression : ECLc ~ FEF:CEF + EPI + COT + Taille + EUR + An2017
##             Estimate Pr(>|t|) star
## (Intercept) -38.1442 0.0105   *
## EPI         -0.1421  3e-04    ***
## COT         -0.4919  0.037    *
## Taille      1.8854   0.1369
## EUR         6.1091   0        ***
## An2017      0.6149   0.2838
## FEF:CEF     0.2769   0        ***
## adj.r.squared =  0.3943181      p.value =  16.08223
## -----------------------------------------
## Regression : ERSEc ~ CEF + EUR + Année + An2017
##             Estimate Pr(>|t|) star
## (Intercept) -33.7664 0        ***
## CEF         1.4942   3e-04    ***
## EUR         2.5156   0        ***
## Année       1.1898   0        ***
## An2017      -0.5485  0.0904   .
## adj.r.squared =  0.5435664      p.value =  42.38375
## -----------------------------------------
## Regression : CDP ~ CEF + EPI + Taille + Croiss + Année + An2018
##             Estimate Pr(>|t|) star
## (Intercept) 1.8609   0.8697
## CEF         1.7355   1e-04    ***
## EPI         0.1388   0        ***
## Taille      -2.5855  0.0024   **
## Croiss      -3.7251  0.1265
## Année       0.4032   0.0194   *
## An2018      0.1287   0.7522
## adj.r.squared =  0.3988257      p.value =  15.48448
## -----------------------------------------
## Regression : ECLc ~ FEF:CEF + EPI + COT + Taille + EUR + An2018
##             Estimate Pr(>|t|) star
## (Intercept) -38.5776 0.0091   **
## EPI         -0.1396  3e-04    ***
## COT         -0.4857  0.0391   *
## Taille      1.9867   0.1104
## EUR         6.032    0        ***
## An2018      -0.6123  0.2709
## FEF:CEF     0.2702   0        ***
## adj.r.squared =  0.394608      p.value =  16.10055
## -----------------------------------------
## Regression : ERSEc ~ CEF + EUR + Année + An2018
##             Estimate Pr(>|t|) star
## (Intercept) -35.4975 0        ***
## CEF         1.5128   3e-04    ***
## EUR         2.5257   0        ***
## Année       1.2746   0        ***
## An2018      -0.3025  0.3527
## adj.r.squared =  0.5367195      p.value =  41.25855
## -----------------------------------------
## Regression : CDP ~ CEF + EPI + Taille + Croiss + Année + An2019
##             Estimate Pr(>|t|) star
## (Intercept) 2.2065   0.8543
## CEF         1.7226   2e-04    ***
## EPI         0.1373   0        ***
## Taille      -2.6024  0.0023   **
## Croiss      -3.6927  0.1342
## Année       0.4109   0.1175
## An2019      -0.0025  0.9968
## adj.r.squared =  0.3983443      p.value =  15.45542
## -----------------------------------------
## Regression : ECLc ~ FEF:CEF + EPI + COT + Taille + EUR + An2019
##             Estimate Pr(>|t|) star
## (Intercept) -43.8352 0.0035   **
## EPI         -0.1171  0.0027   **
## COT         -0.4686  0.0476   *
## Taille      2.3845   0.0601   .
## EUR         5.7189   0        ***
## An2019      0.2351   0.6826
## FEF:CEF     0.2557   0        ***
## adj.r.squared =  0.3898151      p.value =  15.79997
## -----------------------------------------
## Regression : ERSEc ~ CEF + EUR + Année + An2019
##             Estimate Pr(>|t|) star
## (Intercept) -29.9502 0        ***
## CEF         1.4541   4e-04    ***
## EUR         2.4939   0        ***
## Année       0.9737   0        ***
## An2019      0.9051   0.0685   .
## adj.r.squared =  0.5450966      p.value =  42.63984
```

# 13 Régressions Simples

```
names(X)
```

```
##  [1] "X1"       "Année"    "2016"     "2017"     "2018"     "2019"
##  [7] "FEF"      "CEF"      "CDP"      "GRI"      "IV"       "ERSE"
## [13] "ERSEc"    "ECL"      "ECLc"     "COT"      "EUR"      "EPI"
## [19] "Taille"   "Taille.1" "Croiss"   "Renta"
```

# 14 Toutes les régressions dans une boucle automatisée

Les Années ne figurent pas en variable explicative dans ce script.

```
for(y in c('CDP','ECLc','GRI','ERSEc')){
  f <- formula(paste(y,'~FEF*CEF+EPI+COT+Taille+Croiss+Renta+EUR'))
  cat('-----------------------------------------\n')
  cat('Regression : ');print(f)
  reg <- lm(formula = f, data=X)
  print(cbind(summary(reg)$coefficients[,c(1,4)],
              star=star(summary(reg)$coefficients[,c(4)])
              ), quote=FALSE
  )
  cat('adj.r.squared = ',broom::glance(reg)[[2]],'     p.value = ',broom::glance(reg)[['p.value']], '\n
  #plot(reg)
}
```

```
## -----------------------------------------
## Regression : CDP ~ FEF * CEF + EPI + COT + Taille + Croiss + Renta + EUR
##             Estimate            Pr(>|t|)             star
## (Intercept) 416.954420825682    2.75581465206212e-06 ***
## FEF         -40.7056866074756   5.10325711465906e-06 ***
## CEF         -38.9001422590165   2.32250151223598e-05 ***
## EPI         0.0780457797866296  0.00231488961411373  **
## COT         -0.0115645471855912 0.94332733463269
## Taille      -3.35159763130775   0.000114541570295629 ***
## Croiss      -0.657000631155747  0.784853612715777
## Renta       -0.169017984864196  0.763065036981187
## EUR         1.05312312469825    0.0757685895495297   .
## FEF:CEF     4.18838302412443    5.83792043075899e-06 ***
## adj.r.squared =  0.4619849      p.value =  7.922133e-15
## -----------------------------------------
## Regression : ECLc ~ FEF * CEF + EPI + COT + Taille + Croiss + Renta + EUR
##             Estimate            Pr(>|t|)             star
## (Intercept) -62.7445899418481   0.593432674364401
## FEF         1.82195888085493    0.876654230854005
## CEF         3.09164044766154    0.800319608687798
## EPI         -0.141630543192817  0.000156716615650339 ***
## COT         -0.470995655750506  0.0481130773066921   *
## Taille      2.22265843413532    0.0707390842851667   .
## Croiss      -0.14316349263638   0.967210587193914
## Renta       -1.39718044443081   0.0826835932372211   .
## EUR         5.18478394459513    6.4264757721234e-09  ***
## FEF:CEF     0.00623795696082547 0.995912410059061
## adj.r.squared =  0.3912452      p.value =  1.515575e-12
## -----------------------------------------
## Regression : GRI ~ FEF * CEF + EPI + COT + Taille + Croiss + Renta + EUR
##             Estimate             Pr(>|t|)          star
## (Intercept) 130.915335042164     0.53826924071986
## FEF         -11.6366793807856    0.584025721401927
## CEF         -7.42113589874513    0.737268300872779
## EPI         -0.00842470383745052 0.898340208205575
## COT         -0.0994390788574315  0.816302894651302
## Taille      -2.49259787160883    0.260903499272662
## Croiss      0.103383237817586    0.986912218524191
## Renta       0.683328877553211    0.637261930304437
## EUR         1.21659266473709     0.421721242392712
## FEF:CEF     0.953798929668925    0.665225211489881
## adj.r.squared =  -0.03131458      p.value =  0.8498094
## -----------------------------------------
## Regression : ERSEc ~ FEF * CEF + EPI + COT + Taille + Croiss + Renta + EUR
##             Estimate           Pr(>|t|)             star
## (Intercept) 273.428625599711   0.00136860496944672  **
## FEF         -25.2401780892195  0.00302159509482194  **
## CEF         -25.9634929220894  0.00337913460433893  **
## EPI         -0.150526008083359 4.63791204513177e-08 ***
## COT         -0.239794914271453 0.15654820778592
## Taille      -2.48526428723792  0.00496443532719035  **
## Croiss      4.31850222652545   0.0837195581916289   .
## Renta       -0.011350070061559 0.984129925198796
## EUR         5.10037333386347   2.49805949488928e-14 ***
## FEF:CEF     2.81043717662319   0.00149728577968913  **
## adj.r.squared =  0.3639309      p.value =  2.138553e-11
```