Code ▾

Advanced Statistics : Real estate - Boston area

Introduction

- L'immobilier aux US est plutôt une thématique fameuse dans son genre. Notamment au centre de la crise des subprimes, il peut être pertinent de suivre l'évolution du prix du logement aux US (bdd date des années 70)
- Problématique : Quelles variables peuvent expliquer le prix médian d'un bien immobilier dans la banlieue de Boston ?
- Pour ce faire, nous avons grâce au cours de Big Dad et au cours de R développé plusieurs outils : Pour en citer les majeurs, il s'agira d'utiliser la Principal Components Analysis et Backward Stepwise Regression mais aussi Subset Sélection, Cross Validation ou la ridge selection.
- Choix de la variable dépendante : MEDV. Nous avons fait des essais avec les différentes variables, c'est en prenant MEDV comme variables dépendante que le r squared était le plus élevé (74%). De plus, une analyse qualitative des différentes variables nous a permis de soutenir ce choix, l'impact que pouvaient avoir de nombreuses variables sur MEDV étant évident.

Libraries Installation

Hide

Hide

```
install.packages("leaps") # Best Subset Selection
library(leaps)
install.packages("glmnet") # Elastic Net
library(glmnet)
```

Subset Selection Method Code

- Les modèles qui affectent une "pénalité" à l'augmentation du nombre de variables sont plus pertinents. Les autres modèles (RSS et R squared, sans pénalité) indiquent de n'enlever aucun predicteur.
- Les modèles du r^2 ajusté, du BIC et du Cp nous donne le même résultat :

garder le modèle avec 11 variables explicatives.
- Le “significance level” des variables restantes reste proche, sauf pour le predicteur TAX qui devient plus significatif.

Hide

Hide

```
install.packages("leaps") # Best Subset Selection
```

```
essai de l'URL 'https://cran.rstudio.com/bin/macosx/el-capitan/co
ntrib/3.4/leaps_3.0.tgz'
Content type 'application/x-gzip' length 69196 bytes (67 KB)
==================================================
downloaded 67 KB
```

```


The downloaded binary packages are in
    /var/folders/ql/qw81rhln68bcj9f1nhfrqxv40000gn/T//RtmpReI2p1/
downloaded_packages
```

Hide

Hide

```
library(leaps)
#housing<-read.csv("R/Projet R/traitement_housing.csv",sep=";")
#head(housing) ; View(housing) ; print(names(housing)) ; print(di
m(housing))
sum(is.na(housing)) # 0 !
```

```
[1] 0
```

Hide

Hide

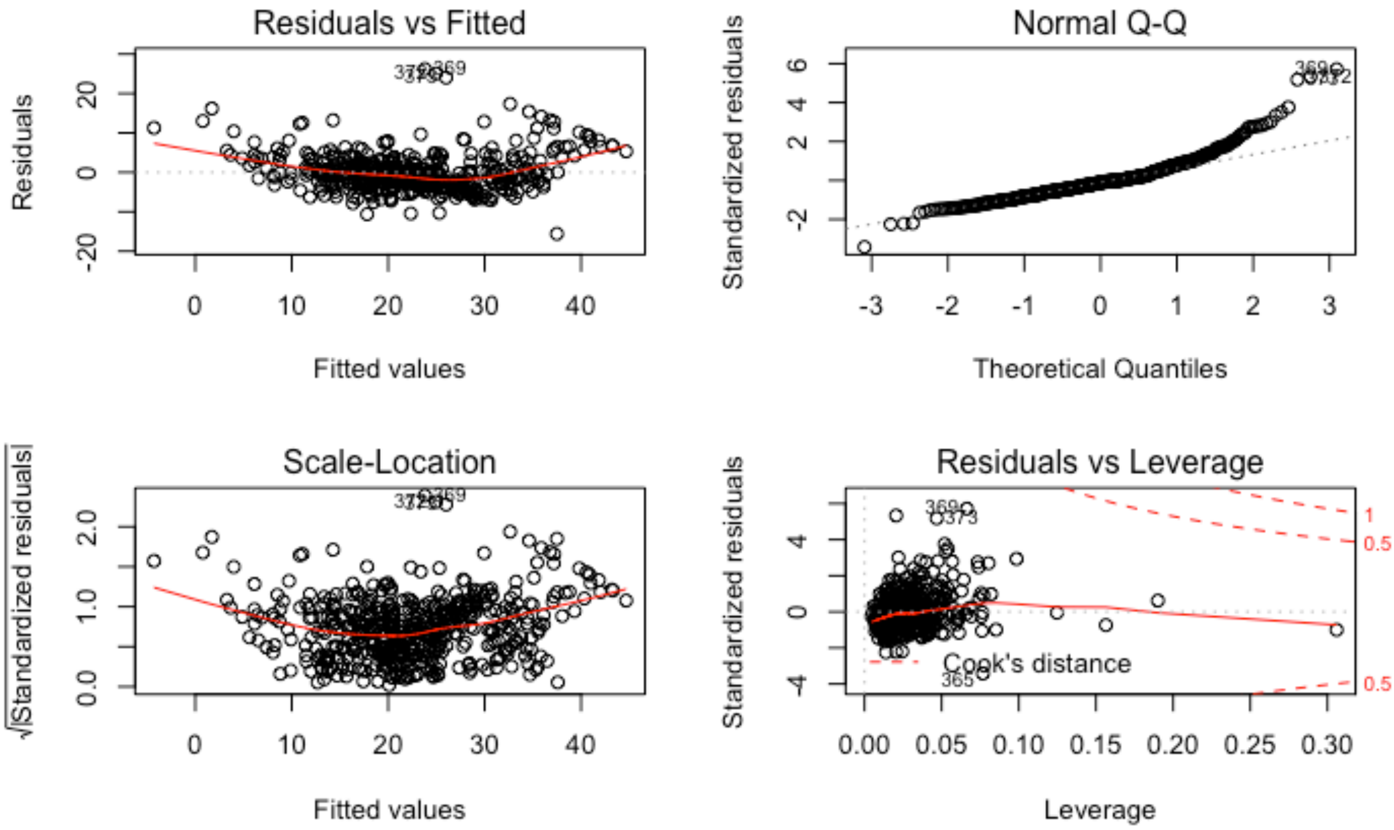
```
reglin<-lm(MEDV~.,housing)
summary(reglin) # r squared of 74% + low pval(Fstat)
```

```
Call:
lm(formula = MEDV ~ ., data = housing)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.595  -2.730  -0.518   1.777  26.199 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
CRIM        -1.080e-01  3.286e-02  -3.287 0.001087 ** 
ZN           4.642e-02  1.373e-02   3.382 0.000778 ***
INDUS        2.056e-02  6.150e-02   0.334 0.738288    
Chase        2.687e+00  8.616e-01   3.118 0.001925 ** 
NOX         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
RM           3.810e+00  4.179e-01   9.116  < 2e-16 ***
AGE          6.922e-04  1.321e-02   0.052 0.958229    
DIS         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
RAD          3.060e-01  6.635e-02   4.613 5.07e-06 ***
TAX         -1.233e-02  3.760e-03  -3.280 0.001112 ** 
PTRATIO     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
B            9.312e-03  2.686e-03   3.467 0.000573 ***
LTSAT       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-squared:  0.7406,    Adjusted R-squared:  0.7338 
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

Hide

Hide

```
par(mfrow=c(2,2))
plot(reglin)
```



Hide

Hide

```
regfit.full<-regsubsets(MEDV~.,housing,nvmax=13)
regsummary=summary(regfit.full)
```

- We can notice that the QQ-plot is not so satisfying. We can therefore try a regression using the log function, we can see that regressing the log of the dependant variable on the logs of the predictors brings a better result as for the QQ-Plot.

Hide

Hide

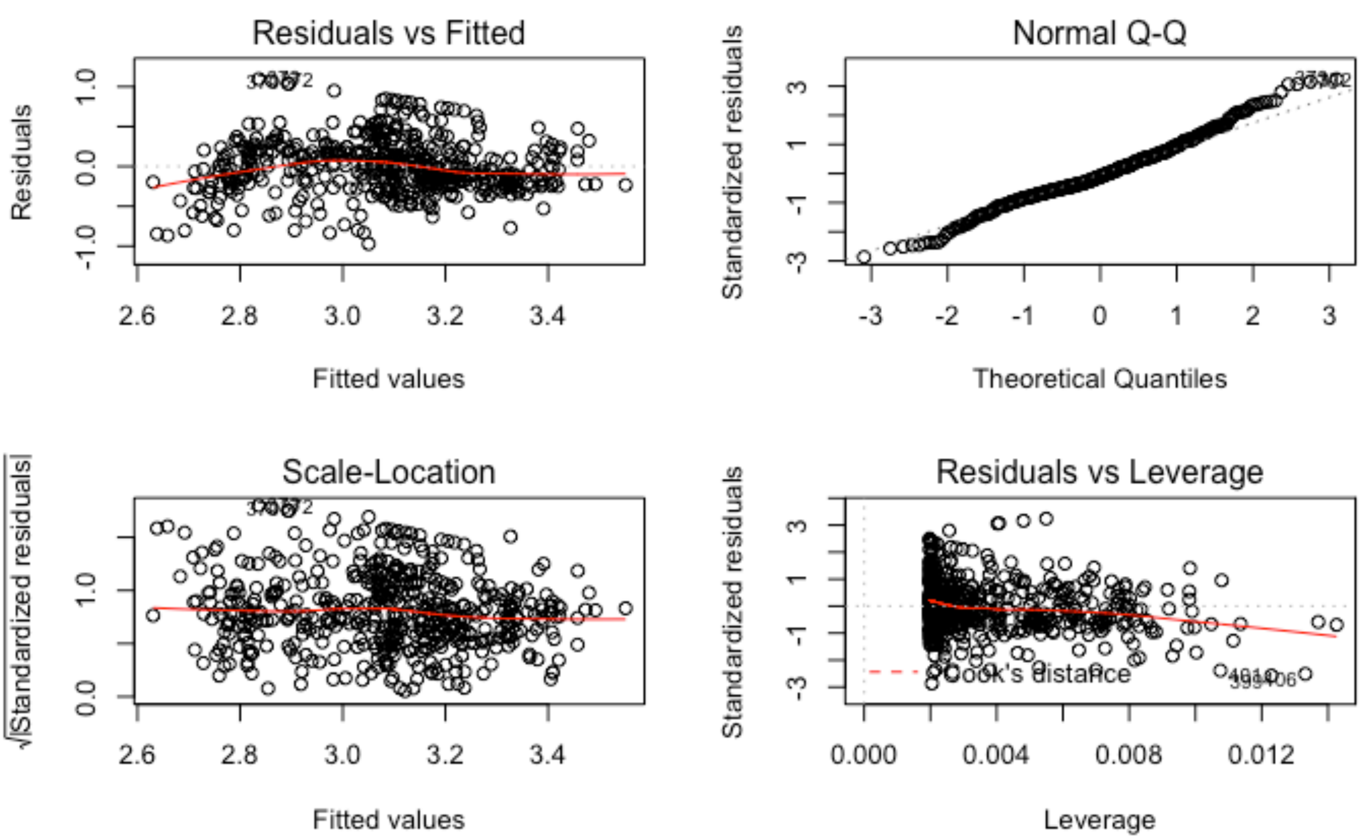
```
predictors<-log(housing$DIS+1)+log(housing$INDUS+1)+log(housing$C
RIM+1)+log(housing$ZN+1)+log(housing$Chase+1)+log(housing$NOX+1)+
log(housing$RM+1)+log(housing$AGE+1)+log(housing$RAD+1)+log(housi
ng$TAX+1)+log(housing$PTRATIO+1)+log(housing$B+1)+log(housing$LTS
AT+1)
logreg <- lm(log(MEDV+1)~predictors,housing)
summary(logreg)
```

```
Call:
lm(formula = log(MEDV + 1) ~ predictors, data = housing)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.97098 -0.20698 -0.03523  0.19437  1.09560 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.541007   0.198846   27.87   <2e-16 ***
predictors  -0.078417   0.006332  -12.38   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3392 on 504 degrees of freedom
Multiple R-squared:  0.2333,    Adjusted R-squared:  0.2318 
F-statistic: 153.4 on 1 and 504 DF,  p-value: < 2.2e-16
```

Hide

Hide

```
par(mfrow=c(2,2))
plot(logreg) # better QQ-plot than with the normal regression...
```



Hide

Hide

Hide

```
# show for each subset size, the variables that should be kept to
have the best model (smallest RSS)
regsummary
```

```
Subset selection object
Call: regsubsets.formula(MEDV ~ ., housing, nvmax = 13)
13 Variables  (and intercept)
        Forced in Forced out
CRIM        FALSE      FALSE
ZN          FALSE      FALSE
INDUS       FALSE      FALSE
Chase       FALSE      FALSE
NOX         FALSE      FALSE
RM          FALSE      FALSE
AGE         FALSE      FALSE
DIS         FALSE      FALSE
RAD         FALSE      FALSE
TAX         FALSE      FALSE
PTRATIO     FALSE      FALSE
B           FALSE      FALSE
LTSAT       FALSE      FALSE
1 subsets of each size up to 13
Selection Algorithm: exhaustive
          CRIM ZN  INDUS Chase NOX RM  AGE DIS RAD TAX PTRATIO B
LTSAT
1  ( 1 )  " "  " " " "   " "   " " " " " " " " " " " " " "     "
" "*"
2  ( 1 )  " "  " " " "   " "   " " "*" " " " " " " " " " "     "
" "*"
3  ( 1 )  " "  " " " "   " "   " " "*" " " " " " " " " "*"     "
" "*"
4  ( 1 )  " "  " " " "   " "   " " "*" " " "*" " " " " "*"     "
" "*"
5  ( 1 )  " "  " " " "   " "   "*" "*" " " "*" " " " " "*"     "
" "*"
6  ( 1 )  " "  " " " "   "*"   "*" "*" " " "*" " " " " "*"     "
" "*"
7  ( 1 )  " "  " " " "   "*"   "*" "*" " " "*" " " " " "*"     "*
" "*"
8  ( 1 )  " "  "*" " "   "*"   "*" "*" " " "*" " " " " "*"     "*
" "*"
9  ( 1 )  "*"  " " " "   "*"   "*" "*" " " "*" "*" " " "*"     "*
```

```
" "*"
10  ( 1 ) "*"  "*" " "  " "  "*" "*" " " "*" "*" "*" "*"    "*
" "*"
11  ( 1 ) "*"  "*" " "  "*"  "*" "*" " " "*" "*" "*" "*"    "*
" "*"
12  ( 1 ) "*"  "*" "*"  "*"  "*" "*" " " "*" "*" "*" "*"    "*
" "*"
13  ( 1 ) "*"  "*" "*"  "*"  "*" "*" "*" "*" "*" "*" "*"    "*
" "*"
```

Hide

Hide

```
names(regsummary) # different methods of selection of the best mo
del between the differents subset sizes
```

```
[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat
" "obj"
```

Hide

Hide

```
par(mfrow=c(3,2))
plot(regsummary$rss,xlab="nb of variables",ylab="RSS",type="l")
plot(regsummary$rsq,xlab="nb of variables",ylab="R squared",type=
"l")
```

Hide

Hide

```
plot(regsummary$adjr2,xlab="nb of variables",ylab="adjusted R squ
ared",type="l")
# no big difference between r2 and adjusted r2
plot(regsummary$cp,xlab="nb of variables",ylab="Cp", type="l")
```

Hide

Hide

```
plot(regsummary$bic,xlab="number of variables",ylab="BIC",type="l
")
which.min(regsummary$rss);which.max(regsummary$rsq)
```

```
[1] 13
[1] 13
```

Hide

Hide

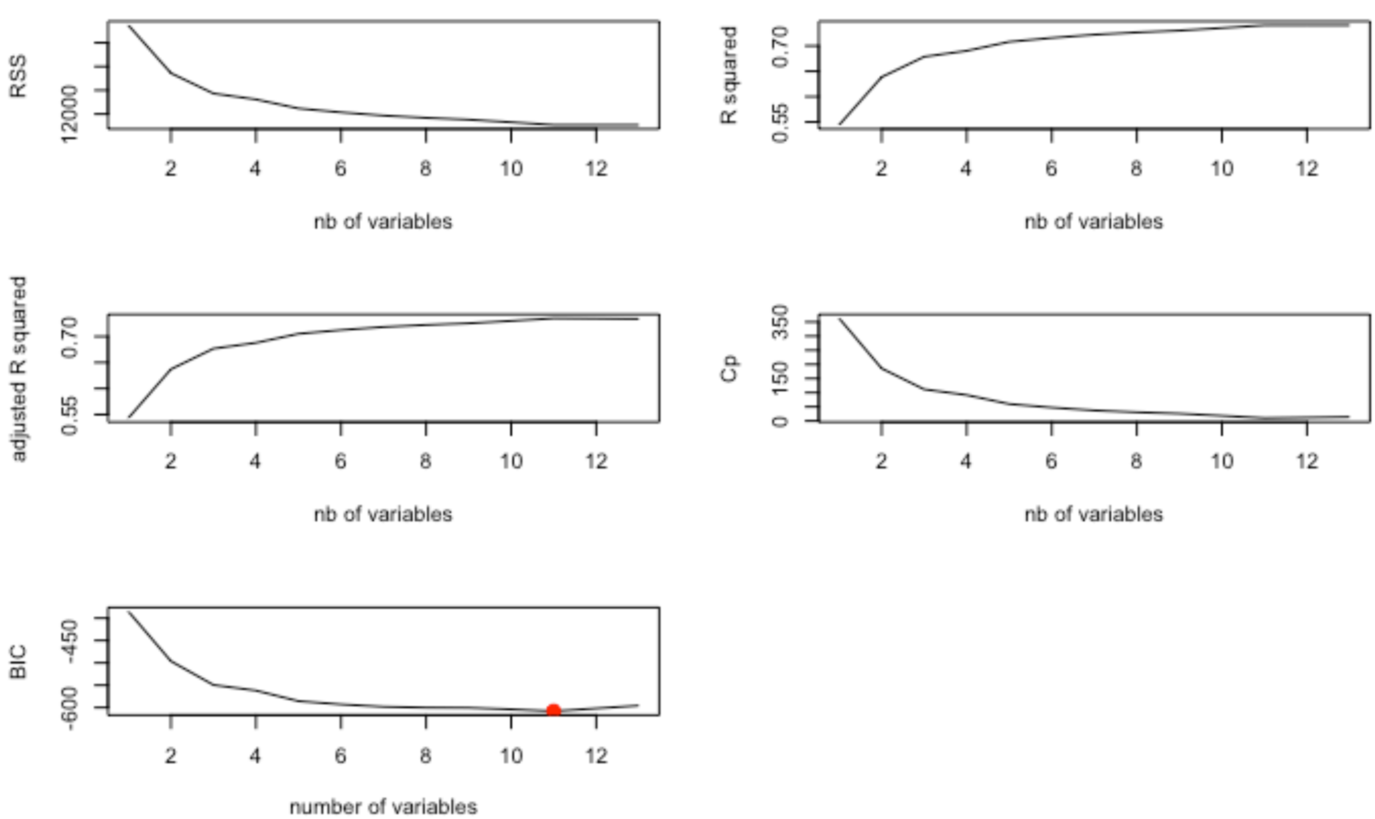
```
# Same result of 13 (the number of explanatory variables) it is c
onsistent with the fact that it always increases/decreases with t
he number of variables
which.max(regsummary$adjr2);which.min(regsummary$cp);which.min(re
gsummary$bic)
```

```
[1] 11
[1] 11
[1] 11
```

Hide

Hide

```
# We obtain the same result (and same as adjr2) : the model with
11 explanatory variables
points(11,regsummary$bic[11],col="red",cex=2,pch=20)
```



Hide

Hide

```
newregfit<-lm(MEDV~CRIM+ZN+Chase+NOX+RM+DIS+RAD+TAX+PTRATIO+B+LTS
AT,housing)
# new regression with the 11 remaining variables
summary(newregfit)
```

```

Call:
lm(formula = MEDV ~ CRIM + ZN + Chase + NOX + RM + DIS + RAD +
    TAX + PTRATIO + B + LTSAT, data = housing)

Residuals:
     Min       1Q   Median       3Q      Max
-15.5984  -2.7386  -0.5046   1.7273  26.2373

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
CRIM         -0.108413   0.032779  -3.307 0.001010 **
ZN            0.045845   0.013523   3.390 0.000754 ***
Chase         2.718716   0.854240   3.183 0.001551 **
NOX         -17.376023   3.535243  -4.915 1.21e-06 ***
RM            3.801579   0.406316   9.356  < 2e-16 ***
DIS          -1.492711   0.185731  -8.037 6.84e-15 ***
RAD           0.299608   0.063402   4.726 3.00e-06 ***
TAX          -0.011778   0.003372  -3.493 0.000521 ***
PTRATIO      -0.946525   0.129066  -7.334 9.24e-13 ***
B             0.009291   0.002674   3.475 0.000557 ***
LTSAT        -0.522553   0.047424 -11.019  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.736 on 494 degrees of freedom
Multiple R-squared:  0.7406,    Adjusted R-squared:  0.7348
F-statistic: 128.2 on 11 and 494 DF,  p-value: < 2.2e-16
```

Hide

Hide

```
summary(lm(MEDV~.,housing)) # just to compare both
```

```
Call:
lm(formula = MEDV ~ ., data = housing)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.595  -2.730  -0.518   1.777  26.199 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
CRIM        -1.080e-01  3.286e-02  -3.287 0.001087 ** 
ZN           4.642e-02  1.373e-02   3.382 0.000778 ***
INDUS        2.056e-02  6.150e-02   0.334 0.738288    
Chase        2.687e+00  8.616e-01   3.118 0.001925 ** 
NOX         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
RM           3.810e+00  4.179e-01   9.116  < 2e-16 ***
AGE          6.922e-04  1.321e-02   0.052 0.958229    
DIS         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
RAD          3.060e-01  6.635e-02   4.613 5.07e-06 ***
TAX         -1.233e-02  3.760e-03  -3.280 0.001112 ** 
PTRATIO     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
B            9.312e-03  2.686e-03   3.467 0.000573 ***
LTSAT       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-squared:  0.7406,    Adjusted R-squared:  0.7338 
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

- While reducing the number of variables, the level of significance of the remaining variables is quite the same as the model with all the predictors, except for the predictor “TAX” which has become more significant
- The multiple r^2 remains unchanged
- The adjusted r^2 slightly increases from 73,38% to 73,48% (linked to the “penalty”)

Hide

Hide

```
coef(regfit.full,11)
```

```
  (Intercept)          CRIM            ZN         Chase 
NOX            RM           DIS 
 36.341145004  -0.108413345   0.045844929   2.718716303 -17.37602
3429   3.801578840  -1.492711460 
          RAD           TAX       PTRATIO             B         L
TSAT 
  0.299608454  -0.011777973  -0.946524570   0.009290845  -0.52255
3457 
```

Forward/Bacward Stepwsise Regression

- La méthode “forward” comme la méthode “backward” nous mènent au même résultat que précédement : le modèle à 11 predictors sans “INDUS” et “AGE”.

Hide

Hide

```
install.packages("leaps")
```

```
essai de l'URL 'https://cran.rstudio.com/bin/macosx/el-capitan/co
ntrib/3.4/leaps_3.0.tgz'
Content type 'application/x-gzip' length 69196 bytes (67 KB)
==================================================
downloaded 67 KB
```

```


The downloaded binary packages are in
	/var/folders/ql/qw81rhln68bcj9f1nhfrqxv40000gn/T//RtmpZKJDqf/
downloaded_packages
```

Hide

Hide

```
library(leaps)
regfit.fwd<-regsubsets(MEDV~.,housing,nvmax=13,method="forward")
summary(regfit.fwd)
```

```
Subset selection object
```

```
Call: regsubsets.formula(MEDV ~ ., housing, nvmax = 13, method =
"forward")
13 Variables  (and intercept)
        Forced in Forced out
CRIM        FALSE      FALSE
ZN          FALSE      FALSE
INDUS       FALSE      FALSE
Chase       FALSE      FALSE
NOX         FALSE      FALSE
RM          FALSE      FALSE
AGE         FALSE      FALSE
DIS         FALSE      FALSE
RAD         FALSE      FALSE
TAX         FALSE      FALSE
PTRATIO     FALSE      FALSE
B           FALSE      FALSE
LTSAT       FALSE      FALSE
1 subsets of each size up to 13
Selection Algorithm: forward
          CRIM ZN  INDUS Chase NOX RM  AGE DIS RAD TAX PTRATIO B
LTSAT
1  ( 1 )  " "  " " " "   " "   " " " " " " " " " " " " " "     "
" "*"
2  ( 1 )  " "  " " " "   " "   " " "*" " " " " " " " " " "     "
" "*"
3  ( 1 )  " "  " " " "   " "   " " "*" " " " " " " " " "*"     "
" "*"
4  ( 1 )  " "  " " " "   " "   " " "*" " " "*" " " " " "*"     "
" "*"
5  ( 1 )  " "  " " " "   " "   "*" "*" " " "*" " " " " "*"     "
" "*"
6  ( 1 )  " "  " " " "   "*"   "*" "*" " " "*" " " " " "*"     "
" "*"
7  ( 1 )  " "  " " " "   "*"   "*" "*" " " "*" " " " " "*"     "*
" "*"
8  ( 1 )  " "  "*" " "   "*"   "*" "*" " " "*" " " " " "*"     "*
" "*"
9  ( 1 )  "*"  "*" " "   "*"   "*" "*" " " "*" " " " " "*"     "*
" "*"
10  ( 1 ) "*"  "*" " "   "*"   "*" "*" " " "*" "*" " " "*"     "*
" "*"
11  ( 1 ) "*"  "*" " "   "*"   "*" "*" " " "*" "*" "*" "*"     "*
" "*"
12  ( 1 ) "*"  "*" "*"   "*"   "*" "*" " " "*" "*" "*" "*"     "*
" "*"
```

```
13  ( 1 ) "*"   "*" "*"   "*"   "*" "*" "*" "*" "*" "*" "*"     "*
" "*"
```

Hide

Hide

```
which.min(summary(regfit.fwd)$bic)
```

```
[1] 11
```

Hide

Hide

```
#regfit.bwd<-regsubsets(MEDV~.,housing,nvmax=13,method="backward"
)
#summary(regfit.bwd)
#which.min(summary(regfit.bwd)$bic)
# We obtain the same result as with the Subset Selection Method
```

Cross-Validation

- Pour des valeurs allant de 2 à 30 pour le nombre de séparations (nombre de “folds”), nous obtenons toujours le même résultat (modèle à 11 variables)

Hide

Hide

```
install.packages("leaps")
```

```
essai de l'URL 'https://cran.rstudio.com/bin/macosx/el-capitan/co
ntrib/3.4/leaps_3.0.tgz'
Content type 'application/x-gzip' length 69196 bytes (67 KB)
==================================================
downloaded 67 KB
```

```


The downloaded binary packages are in
	/var/folders/ql/qw81rhln68bcj9f1nhfrqxv40000gn/T//Rtmp8NXX3o/
downloaded_packages
```

Hide

Hide

Hide

```
library(leaps)
regfit.best=regsubsets(MEDV~.,data=housing,nvmax=13)
# ## c.v.
k=15 # ou autre ...
for(k in 1:30)
set.seed(8)
# create a vector that allocates each obs to one of the k=15 fold
s
folds=sample(1:k,nrow(housing),replace=TRUE)
cv.errors=matrix(NA,k,13, dimnames=list(NULL, paste(1:13)))
for(j in 1:k){
  best.fit=regsubsets(MEDV~.,data=housing[folds!=j,],nvmax=13) #
estimate outside the fold j
  for(i in 1:13){
    pred=predict(best.fit,housing[folds==j,],id=i)
    cv.errors[j,i]=mean((housing$MEDV[folds==j]-pred)^2)
  }
}
mean.cv.errors=apply(cv.errors,2,mean)
# mean across 13 models (k=15 folds)
mean.cv.errors
```

```
       1        2        3        4        5        6        7
8        9       10       11       12
38.71550 31.80783 28.23559 28.49333 25.83182 26.78187 25.52970 25
.84712 26.42541 25.90942 23.87513 24.02937
      13
24.10530
```

Hide

Hide

```
which.min(mean.cv.errors)
```

```
11
11
```

Hide

Hide

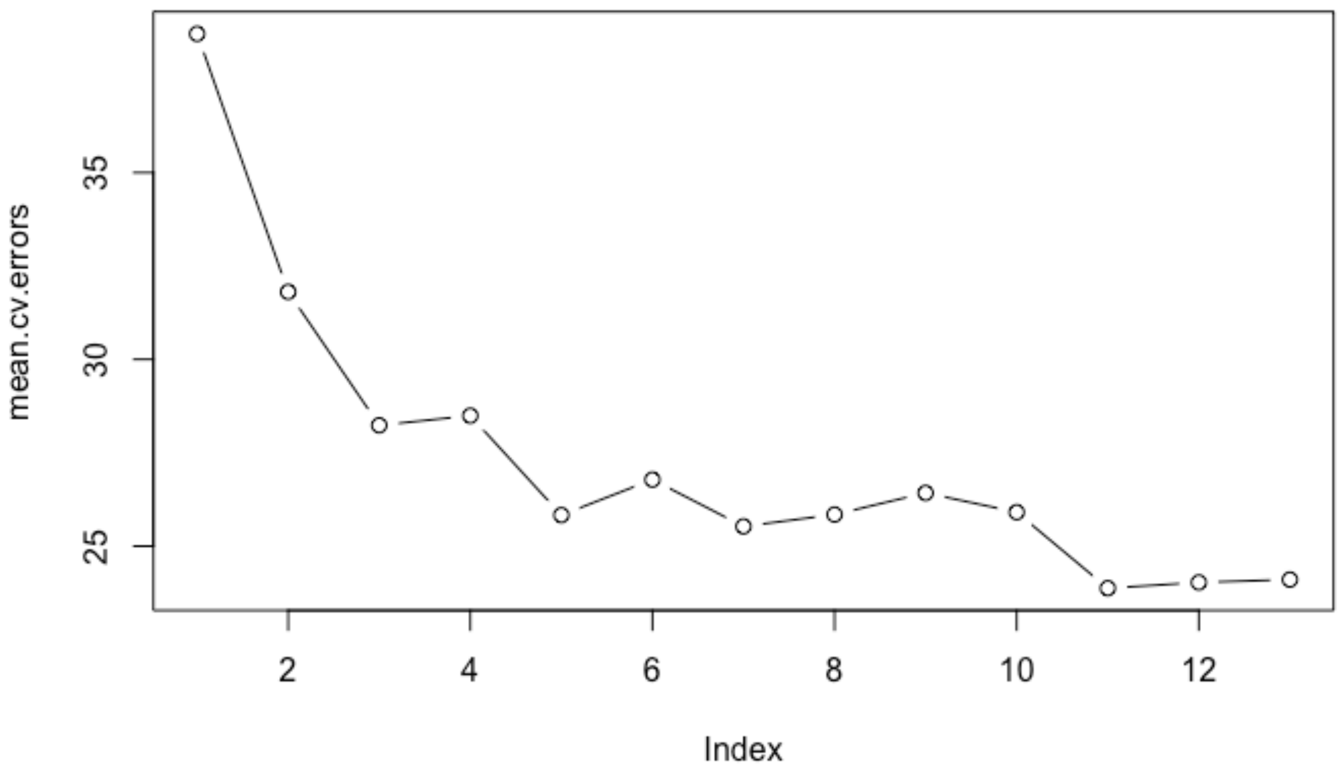
```
par(mfrow=c(1,1))
```

- Using k=10, the model with 11 predictors should be kept according to this method, however the different means (of the SSR values obtain in the 10 subsets) are very close, therefore we could reasonably think that using a penalty, we would obtain a lower number of predictors.

Hide

Hide

```
plot(mean.cv.errors,type='b')
```



Hide

Hide

```
reg.best=regsubsets(MEDV~.,data=housing, nvmax=13)
coef(reg.best,11)
```

```
  (Intercept)          CRIM             ZN         Chase
NOX            RM           DIS
 36.341145004  -0.108413345   0.045844929   2.718716303 -17.37602
3429   3.801578840  -1.492711460
         RAD           TAX       PTRATIO             B         L
TSAT
  0.299608454  -0.011777973  -0.946524570   0.009290845  -0.52255
3457
```

Ridge Regression Code

- Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity
- By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It is hoped that the net effect will be to give estimates that are more reliable.
- If the number of regressors is larger than the number of data points (deg of liberty = n-p-1 < 0) all regresors cannot be included in the regression.

Hide

Hide

```
install.packages("glmnet")
```

```
essai de l'URL 'https://cran.rstudio.com/bin/macosx/el-capitan/co
ntrib/3.4/glmnet_2.0-16.tgz'
Content type 'application/x-gzip' length 1485951 bytes (1.4 MB)
==================================================
downloaded 1.4 MB
```

```


The downloaded binary packages are in
	/var/folders/ql/qw81rhln68bcj9f1nhfrqxv40000gn/T//Rtmp8NXX3o/
downloaded_packages
```

Hide

Hide

```
library(glmnet)
```

```
le package ‘glmnet’ a été compilé avec la version R 3.4.4Le charg
ement a nécessité le package : Matrix
Le chargement a nécessité le package : foreach
Loaded glmnet 2.0-16
```

Hide

Hide

```
# package for elastic net
x=model.matrix(MEDV~.,housing)[,-1]
y=housing$MEDV
grid=10^seq(10,-2,length=121)
grid
```

```
  [1] 1.000000e+10 7.943282e+09 6.309573e+09 5.011872e+09 3.98107
2e+09 3.162278e+09 2.511886e+09
  [8] 1.995262e+09 1.584893e+09 1.258925e+09 1.000000e+09 7.94328
2e+08 6.309573e+08 5.011872e+08
 [15] 3.981072e+08 3.162278e+08 2.511886e+08 1.995262e+08 1.58489
3e+08 1.258925e+08 1.000000e+08
 [22] 7.943282e+07 6.309573e+07 5.011872e+07 3.981072e+07 3.16227
8e+07 2.511886e+07 1.995262e+07
 [29] 1.584893e+07 1.258925e+07 1.000000e+07 7.943282e+06 6.30957
3e+06 5.011872e+06 3.981072e+06
 [36] 3.162278e+06 2.511886e+06 1.995262e+06 1.584893e+06 1.25892
5e+06 1.000000e+06 7.943282e+05
 [43] 6.309573e+05 5.011872e+05 3.981072e+05 3.162278e+05 2.51188
6e+05 1.995262e+05 1.584893e+05
 [50] 1.258925e+05 1.000000e+05 7.943282e+04 6.309573e+04 5.01187
2e+04 3.981072e+04 3.162278e+04
 [57] 2.511886e+04 1.995262e+04 1.584893e+04 1.258925e+04 1.00000
0e+04 7.943282e+03 6.309573e+03
 [64] 5.011872e+03 3.981072e+03 3.162278e+03 2.511886e+03 1.99526
2e+03 1.584893e+03 1.258925e+03
 [71] 1.000000e+03 7.943282e+02 6.309573e+02 5.011872e+02 3.98107
2e+02 3.162278e+02 2.511886e+02
 [78] 1.995262e+02 1.584893e+02 1.258925e+02 1.000000e+02 7.94328
2e+01 6.309573e+01 5.011872e+01
 [85] 3.981072e+01 3.162278e+01 2.511886e+01 1.995262e+01 1.58489
3e+01 1.258925e+01 1.000000e+01
 [92] 7.943282e+00 6.309573e+00 5.011872e+00 3.981072e+00 3.16227
8e+00 2.511886e+00 1.995262e+00
 [99] 1.584893e+00 1.258925e+00 1.000000e+00 7.943282e-01 6.30957
3e-01 5.011872e-01 3.981072e-01
[106] 3.162278e-01 2.511886e-01 1.995262e-01 1.584893e-01 1.25892
5e-01 1.000000e-01 7.943282e-02
[113] 6.309573e-02 5.011872e-02 3.981072e-02 3.162278e-02 2.51188
6e-02 1.995262e-02 1.584893e-02
[120] 1.258925e-02 1.000000e-02
```

Hide

Hide

```
ridge.mod=glmnet(x,y,alpha=0,lambda=grid)
dim(coef(ridge.mod))
```

```
[1]  14 121
```

Hide

Hide

```
ridge.mod$lambda[50]
```

```
[1] 125892.5
```

Hide

Hide

```
coef(ridge.mod)[,50]
```

```
  (Intercept)          CRIM            ZN         INDUS         C
hase           NOX            RM 
 2.253476e+01 -3.029073e-05  1.036971e-05 -4.730956e-05  4.631163
e-04 -2.474185e-03  6.641561e-04 
          AGE           DIS           RAD           TAX         PTR
ATIO             B         LTSAT 
-8.984429e-06  7.961041e-05 -2.940414e-05 -1.865180e-06 -1.573893
e-04  2.450736e-06 -6.931583e-05 
```

Hide

Hide

```
sqrt(sum(coef(ridge.mod)[-1,50]^2))
```

```
[1] 0.002610996
```

Hide

Hide

```
predict(ridge.mod,s=50,type="coefficients")[1:14,]
```

```
 (Intercept)         CRIM           ZN        INDUS        Chase
NOX          RM          AGE
23.599833103 -0.036202778  0.011669706 -0.052515834  0.904518380
-2.578718769  1.124746737 -0.008817626
         DIS          RAD          TAX      PTRATIO            B
LTSAT
 0.023153757 -0.026417452 -0.002035220 -0.236342882  0.003131382
-0.105893106
```

Hide

Hide

```
# Train and validate
set.seed(121)
train=sample(1:nrow(x),nrow(x)/2) # 2 datasets of same size
test=(-train)
y.test=y[test]
# first try with lambda=4
ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=grid,thresh=1e
-12) # we calibrate on the "train" set of data
ridge.pred=predict(ridge.mod,s=4,newx=x[test,]) # we predict on t
he "test" set using the "train" calibration to predict
mean((ridge.pred-y.test)^2) # out of sample error
```

```
[1] 31.03405
```

Hide

Hide

```
mean((mean(y[train])-y.test)^2) # error relative to mean
```

```
[1] 79.80175
```

Hide

Hide

```
# then we can try with very high lambda, as lambda increases, the
coefficient are drived toward zero, therefore the dependant varia
ble should be close to the mean
ridge.pred=predict(ridge.mod,s=1e10,newx=x[test,])
mean((ridge.pred-y.test)^2)
```

```
[1] 79.80175
```

- As forecasted, it is equal to the precedent result. Indeed, when lambda is very high (~infinite penalty), the betas of the regresssion calibrated on “train” are close to 0, so the predicted variables are merely equals to the intercept, meaning the mean of the “train” dataset. In other words : ridge.pred=mean(y[train])

Hide

Hide

```
ridge.pred=predict(ridge.mod,s=0,newx=x,x=x,y=y,exact=T)
mean((ridge.pred-y.test)^2)
```

```
[1] 161.321
```

Hide

Hide

```
lm(y~x,subset=train)
```

```


Call:
lm(formula = y ~ x, subset = train)

Coefficients:
(Intercept)        xCRIM          xZN       xINDUS       xChase 
xNOX          xRM         xAGE  
  32.135256    -0.129002     0.057459    -0.036471     3.536340 
-15.429422     4.159140    -0.015128  
       xDIS         xRAD         xTAX     xPTRATIO           xB 
xLTSAT  
  -1.570639     0.213266    -0.008480    -0.870637     0.008291 
-0.467188  
```

Hide

Hide

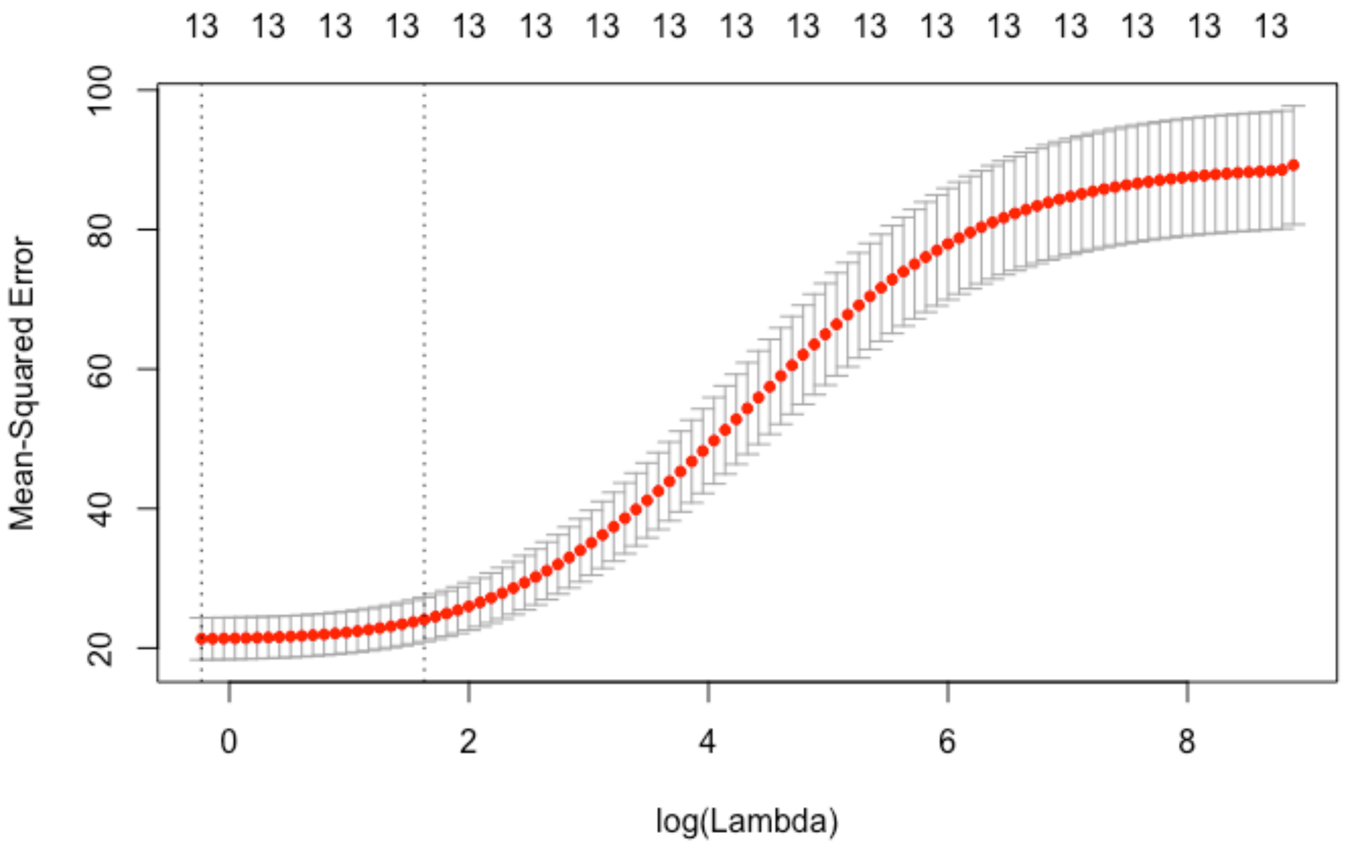
```
predict(ridge.mod,s=0,exact=T,x=x,y=y,type="coefficients")[1:14,]
```

```
  (Intercept)          CRIM            ZN         INDUS          C
hase          NOX           RM
 36.459353996  -0.108010484   0.046420321   0.020556585   2.68674
0920 -17.766522516   3.809870036
         AGE           DIS           RAD           TAX         PTR
ATIO             B        LTSAT
  0.000692156  -1.475566864   0.306044516  -0.012334382  -0.95274
4778   0.009311670  -0.524758345
```

Hide

Hide

```
# With the Cross Validation Method
set.seed(121)
cv.out=cv.glmnet(x[train,],y[train],alpha=0) # 10-fold CV by defa
ult
plot(cv.out) #log !
```



Hide

Hide

```
bestlambda=cv.out$lambda.min
bestlambda # 0.79
```

```
[1] 0.7920875
```

Hide

Hide

```
ridge.pred=predict(ridge.mod,s=bestlambda,newx=x[test,])
mean((ridge.pred-y.test)^2)
```

```
[1] 28.27471
```

Hide

Hide

```
out=glmnet(x,y,alpha=0)
predict(out,type="coefficients",s=bestlambda)[1:14,]
```

```
  (Intercept)          CRIM             ZN          INDUS           C
hase           NOX            RM
 27.212796997  -0.085780668   0.031448692  -0.041809644   2.90994
6108 -11.319916884   4.017693673
          AGE           DIS            RAD            TAX         PTR
ATIO             B         LTSAT
 -0.004104177  -1.076112442   0.141953232  -0.005342391  -0.84374
8157   0.009023398  -0.465463708
```

- We can notice that most betas decrease in absolute value except INDUS, CHASE, RM and AGE. However, INDUS and AGE were already very low with the OLS regression, so we could suggest that CHASE and LM have a really significant impact on MEDV. (they withstand the ridge effect …)

PC ANALYSIS

Loading de la base de données et traitement de base

Hide

Hide

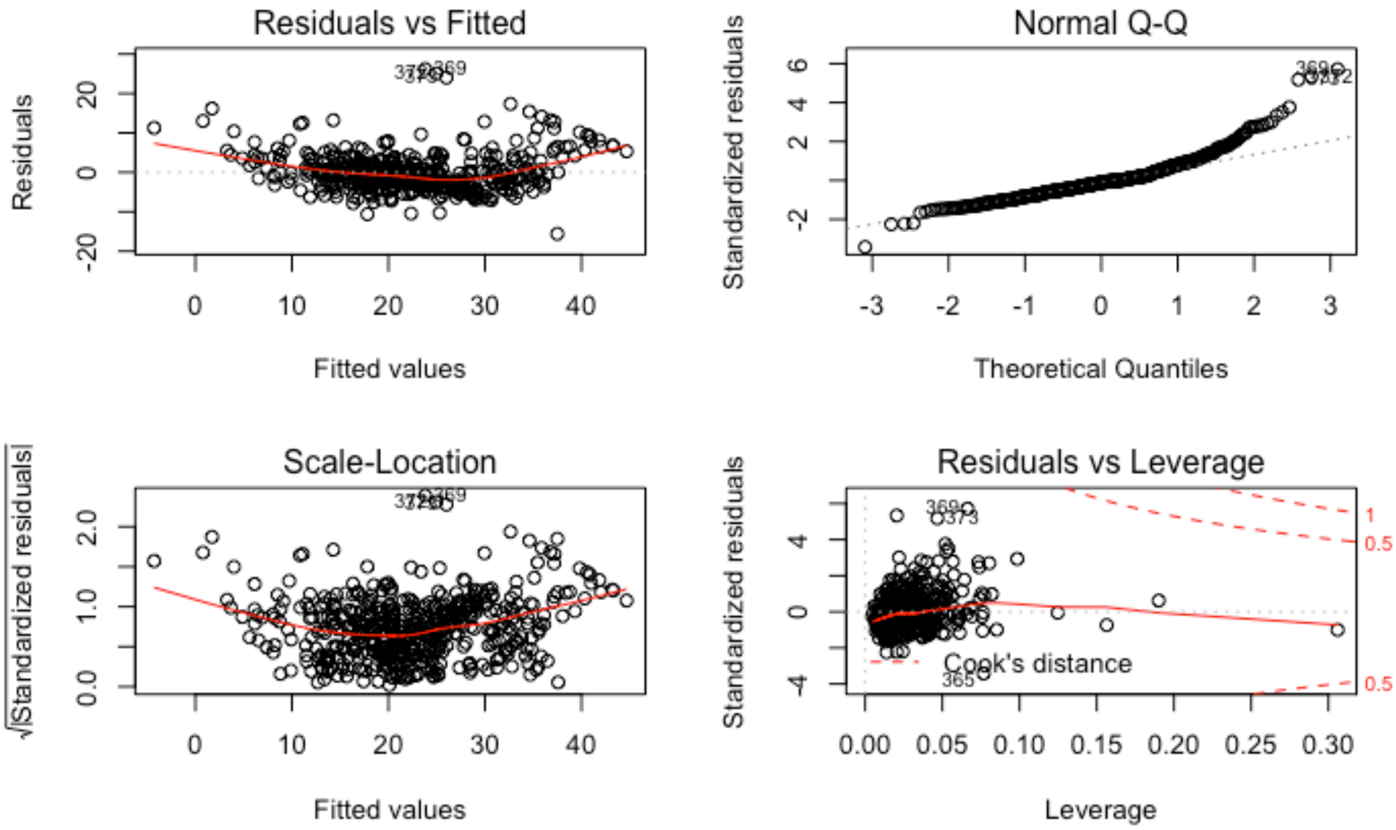
```
#housingdataset=read.csv("C:/Users/alexg/Desktop/Informatique/R/projet/traitement_housing.csv")
regression_initiale=lm(housing$MEDV~.,housing)
summary(regression_initiale)
```

```
Call:
lm(formula = housing$MEDV ~ ., data = housing)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.595  -2.730  -0.518   1.777  26.199 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
CRIM        -1.080e-01  3.286e-02  -3.287 0.001087 ** 
ZN           4.642e-02  1.373e-02   3.382 0.000778 ***
INDUS        2.056e-02  6.150e-02   0.334 0.738288    
Chase        2.687e+00  8.616e-01   3.118 0.001925 ** 
NOX         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
RM           3.810e+00  4.179e-01   9.116  < 2e-16 ***
AGE          6.922e-04  1.321e-02   0.052 0.958229    
DIS         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
RAD          3.060e-01  6.635e-02   4.613 5.07e-06 ***
TAX         -1.233e-02  3.760e-03  -3.280 0.001112 ** 
PTRATIO     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
B            9.312e-03  2.686e-03   3.467 0.000573 ***
LTSAT       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-squared:  0.7406,    Adjusted R-squared:  0.7338 
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

Hide

Hide

```
par(mfrow=c(2,2))
plot(regression_initiale)
```

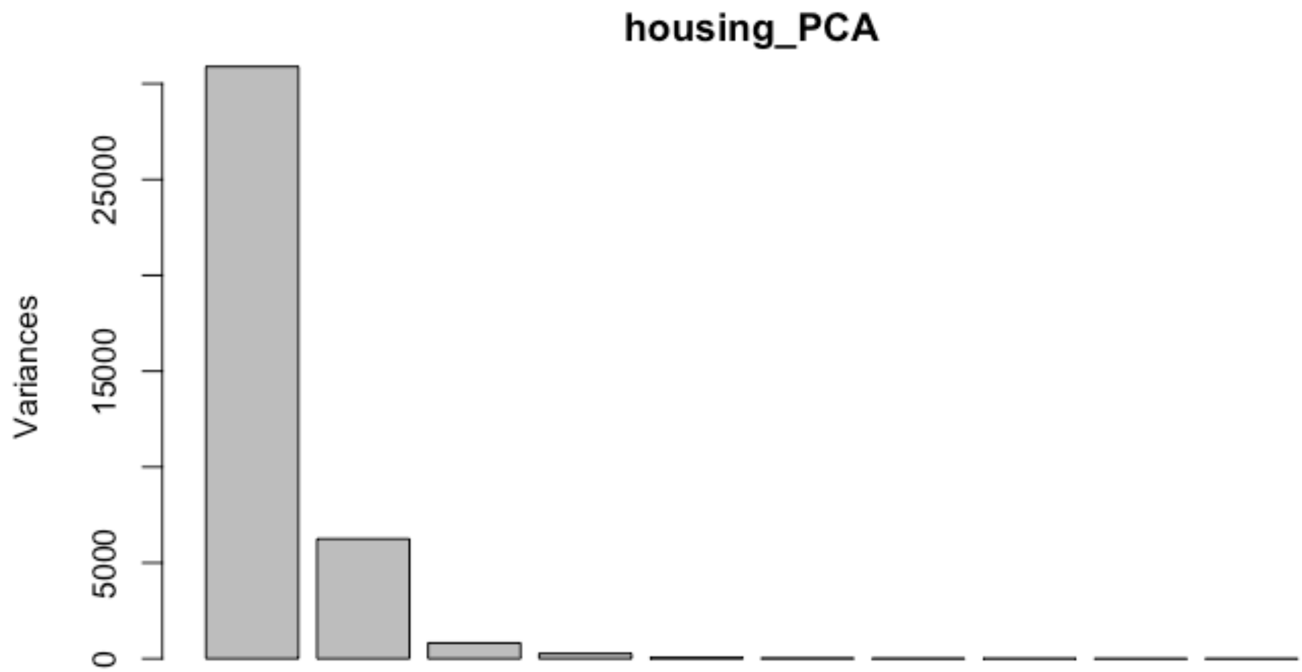


PC Creation

Hide

Hide

```
housing_PCA=prcomp(housing[,-14])
plot(housing_PCA)
```



Hide

Hide

```
summary(housing_PCA)
```

```
Importance of components:
                            PC1     PC2      PC3      PC4    PC5
PC6     PC7     PC8     PC9    PC10
Standard deviation     175.7553 79.0590 28.60706 16.33049 7.0591
5.27985 4.00792 3.08664 1.80924 1.08671
Proportion of Variance   0.8058  0.1631  0.02135  0.00696 0.0013
0.00073 0.00042 0.00025 0.00009 0.00003
Cumulative Proportion    0.8058  0.9689  0.99022  0.99718 0.9985
0.99921 0.99963 0.99988 0.99996 0.99999
                          PC11   PC12    PC13
Standard deviation     0.50513 0.2451 0.05527
Proportion of Variance 0.00001 0.0000 0.00000
Cumulative Proportion  1.00000 1.0000 1.00000
```

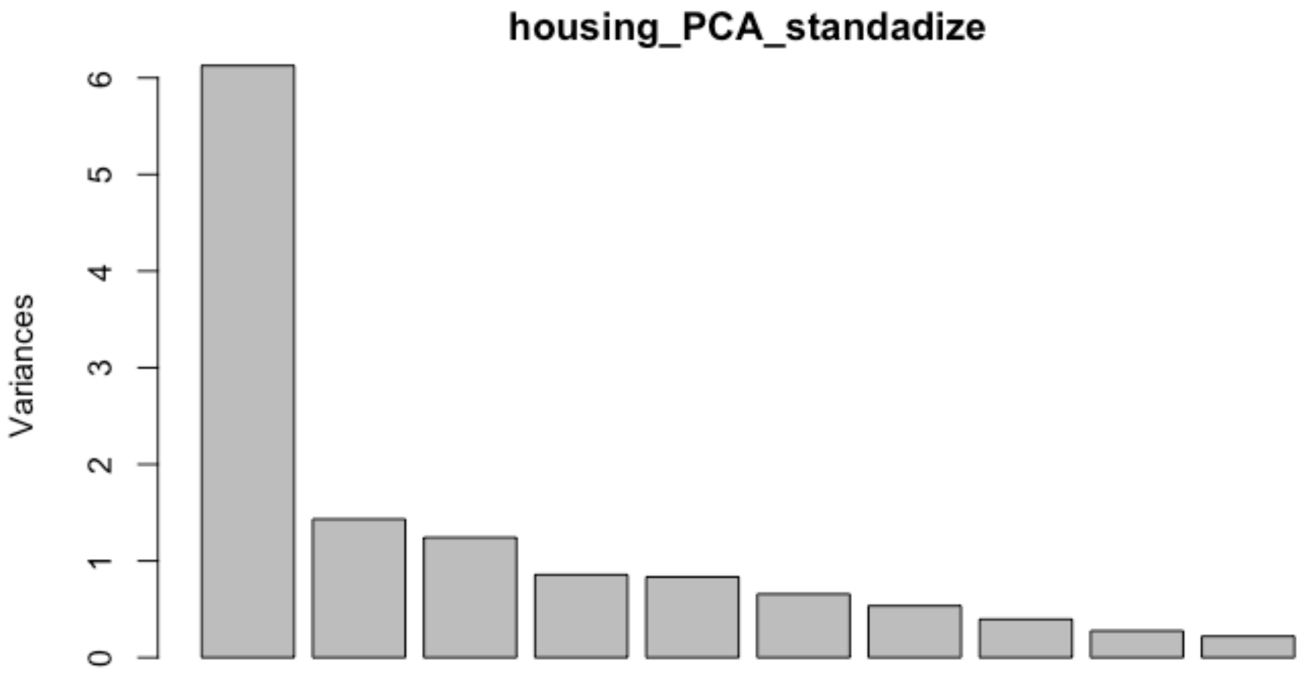
- Par la suite,si on standardise les données afin d’éviter les problèmes de grandes différences de variances entre les variables explicatives on obtient :

Correction PC standardisées

Hide

Hide

```
housing_PCA_standadize=prcomp(housing[,-14],scale=T)
plot(housing_PCA_standadize)
```



Hide

Hide

```
summary(housing_PCA_standadize)
```

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5
PC6     PC7     PC8    PC9    PC10    PC11
Standard deviation     2.4752 1.1972 1.11473 0.92605 0.91368 0.81
081 0.73168 0.62936 0.5263 0.46930 0.43129
Proportion of Variance 0.4713 0.1103 0.09559 0.06597 0.06422 0.05
057 0.04118 0.03047 0.0213 0.01694 0.01431
Cumulative Proportion  0.4713 0.5816 0.67713 0.74310 0.80732 0.85
789 0.89907 0.92954 0.9508 0.96778 0.98209
                          PC12    PC13
Standard deviation     0.41146 0.25201
Proportion of Variance 0.01302 0.00489
Cumulative Proportion  0.99511 1.00000
```

Corrélation entre les valeurs des PC et MEDv

Hide

Hide

```
cor(housing[,14],housing_PCA_standadize$x)
```

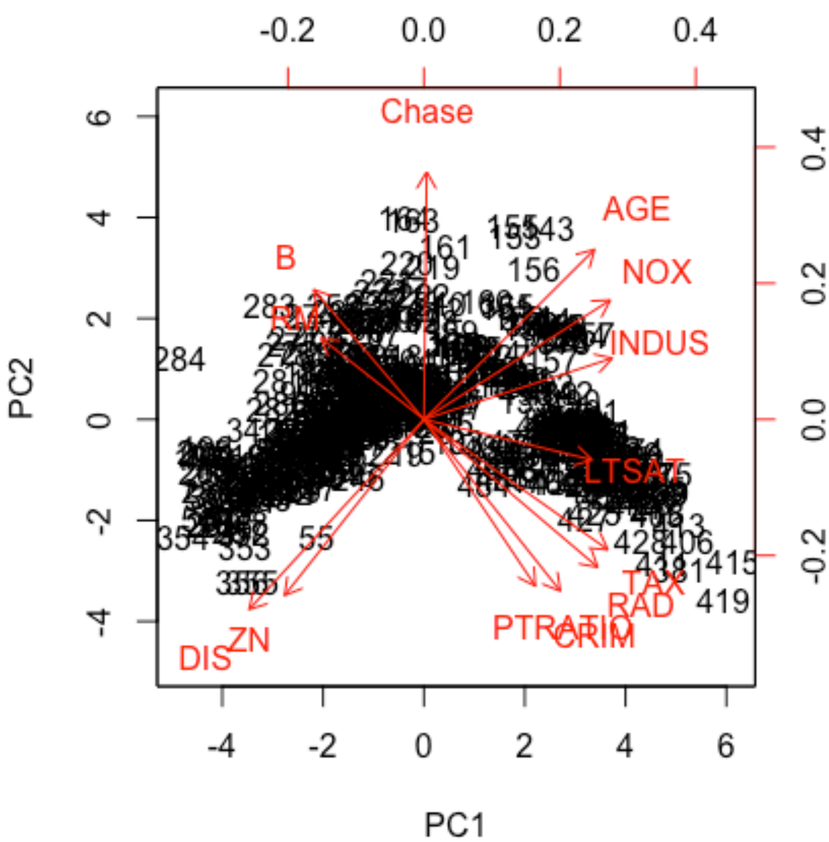
```
              PC1        PC2        PC3        PC4         PC5
PC6            PC7           PC8          PC9
[1,] -0.6117451 0.2857137 0.4243341 0.1088137 -0.2218448 -0.05912
219 -0.007503243 -0.07118018 0.008551394
              PC10        PC11       PC12         PC13
[1,] -0.05657239 0.06441735 0.1379644 -0.09266911
```

Analyse PCA et de la relation entre les variables de bases et les PC1,PC2

Hide

Hide

```
biplot(housing_PCA_standadize,scale=0)
```



- Grace au biplot, on voit comment les PC1&2 sont influencées par les variables explicatives initiales. Notament on voit que l'age,NOX,INDUS augmentent la valeur de la PC1 de façon importante alors que DIS,ZN la diminuent et que Chase a un impact très réduit.

Construction et regression linéaire sur MEDV en fonction de toutes les PCs

Hide

Hide

```
newbase_13=cbind(housing_PCA_standadize$x,housing["MEDV"])
newbase_13
```

PC1 <dbl>	PC2 <dbl>	PC3 <dbl>	PC4 <dbl>	<
-2.0962230302	0.772348426	0.342603683	0.890892398	0.422652
-1.4558109894	0.591399952	-0.694512011	0.486976617	-0.195682
-2.0725465519	0.599046578	0.166956375	0.738473392	-0.933610
-2.6089217589	-0.006863826	-0.100184990	0.343381425	-1.103863
-2.4557547719	0.097615346	-0.075273718	0.427483833	-1.064870
-2.2126618432	-0.009477633	-0.671716355	0.175736244	-0.626567
-1.3575376559	0.349526292	-0.371631528	0.397740214	1.072086
-0.8412121417	0.577228493	-0.518027787	0.537226588	1.378324
-0.1797503956	0.342179528	-1.348304420	0.245676894	2.347980
-1.0731221380	0.315888821	-0.557917054	0.379556648	1.429116

1-10 of 506 rows | 1-7 of 14... Previous **1** 2 3 4 5 6 ... 51 Next

Hide

Hide

```
regression_newbase13=lm(MEDV~.,newbase_13)
regression_newbase13
```

```
Call:
lm(formula = MEDV ~ ., data = newbase_13)

Coefficients:
(Intercept)          PC1          PC2          PC3          PC4          
PC5          PC6          PC7  
   22.53281     -2.27302      2.19491      3.50099      1.08068     
-2.23308     -0.67063     -0.09431  
        PC8          PC9         PC10         PC11         PC12         
PC13  
   -1.04018      0.14945     -1.10869      1.37366      3.08380     
-3.38195  
```

Hide

Hide

```
summary(regression_newbase13)
```

```
Call:
lm(formula = MEDV ~ ., data = newbase_13)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.595  -2.730  -0.518   1.777  26.199 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 22.53281    0.21095 106.814  < 2e-16 ***
PC1         -2.27302    0.08531 -26.644  < 2e-16 ***
PC2          2.19491    0.17638  12.444  < 2e-16 ***
PC3          3.50099    0.18943  18.482  < 2e-16 ***
PC4          1.08068    0.22802   4.739 2.81e-06 ***
PC5         -2.23308    0.23111  -9.662  < 2e-16 ***
PC6         -0.67063    0.26044  -2.575  0.01031 *  
PC7         -0.09431    0.28860  -0.327  0.74396    
PC8         -1.04018    0.33552  -3.100  0.00204 ** 
PC9          0.14945    0.40126   0.372  0.70972    
PC10        -1.10869    0.44996  -2.464  0.01408 *  
PC11         1.37366    0.48960   2.806  0.00522 ** 
PC12         3.08380    0.51320   6.009 3.64e-09 ***
PC13        -3.38195    0.83791  -4.036 6.30e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-squared:  0.7406,	Adjusted R-squared:  0.7338 
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

- Nous venons de régresser MEDV en fonctionn de toutes les PC.
- Il s'agira en suite de déterminer quels PC garder et lesquels supprimer.

Construction et regression linéaire avec les pc de 1 à 5 (cf rule of thumb cours big dad)

- on choisit les 5 premières PC tels que la variance totale du modèle est expliquée à au moins 80%

Hide

Hide

```
newbase_5=cbind(housing_PCA_standadize$x[,1:5],housing["MEDV"])
newbase_5
```

PC1	PC2	PC3	PC4	
<dbl>	<dbl>	<dbl>	<dbl>	<
-2.0962230302	0.772348426	0.342603683	0.890892398	0.422652
-1.4558109894	0.591399952	-0.694512011	0.486976617	-0.195682
-2.0725465519	0.599046578	0.166956375	0.738473392	-0.933610
-2.6089217589	-0.006863826	-0.100184990	0.343381425	-1.103863
-2.4557547719	0.097615346	-0.075273718	0.427483833	-1.064870
-2.2126618432	-0.009477633	-0.671716355	0.175736244	-0.626567
-1.3575376559	0.349526292	-0.371631528	0.397740214	1.072086
-0.8412121417	0.577228493	-0.518027787	0.537226588	1.378324
-0.1797503956	0.342179528	-1.348304420	0.245676894	2.347980
-1.0731221380	0.315888821	-0.557917054	0.379556648	1.429116

1-10 of 506 rows   Previous **1** 2 3 4 5 6 ... 51 Next

Hide

Hide

```
regression_newbase5=lm(MEDV~PC1+PC2+PC3+PC4+PC5,newbase_5)
regression_newbase5
```

```


Call:
lm(formula = MEDV ~ PC1 + PC2 + PC3 + PC4 + PC5, data = newbase_5
)

Coefficients:
(Intercept)          PC1          PC2          PC3          PC4
PC5
     22.533       -2.273        2.195        3.501        1.081
-2.233
```

Hide

Hide

```
summary(regression_newbase5)
```

```


Call:
lm(formula = MEDV ~ PC1 + PC2 + PC3 + PC4 + PC5, data = newbase_5
)

Residuals:
    Min      1Q  Median      3Q     Max 
-19.761  -2.893  -0.758   1.728  33.098 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 22.53281    0.22619  99.619  < 2e-16 ***
PC1         -2.27302    0.09147 -24.850  < 2e-16 ***
PC2          2.19491    0.18912  11.606  < 2e-16 ***
PC3          3.50099    0.20311  17.237  < 2e-16 ***
PC4          1.08068    0.24449   4.420 1.21e-05 ***
PC5         -2.23308    0.24780  -9.012  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.088 on 500 degrees of freedom
Multiple R-squared:  0.697,	Adjusted R-squared:  0.6939 
F-statistic:   230 on 5 and 500 DF,  p-value: < 2.2e-16
```

- Avec seulement les 5 premières variables on a réussi à obtenir un R**2=0.7 et toutes les variables sont significatives.

Corrélation entre les PCs dont la pvalue est infèrieure à 1%

Hide

Hide

```
newbase_4=cbind(housing_PCA_standadize$x[,c(1,2,3,4,5,8,11,12,13)
],housing["MEDV"])
cor(newbase_4[,-10],housing[,-14])
```

```
             CRIM          ZN         INDUS        Chase           NO
X           RM          AGE          DIS
PC1   0.62116675 -0.63444186  0.858099068  0.012481272  0.8486442
4 -0.46842215  0.776412273 -0.795900561
PC2  -0.37741846 -0.38706849  0.134675912  0.544519026  0.2623239
5  0.17877893  0.373498139 -0.417904755
PC3   0.27485437  0.32980075 -0.017775345  0.323026549  0.1348419
8  0.66210465 -0.019702590 -0.055442375
PC4  -0.05720298 -0.11919382 -0.015877849 -0.755605366  0.1187442
7  0.25984306  0.162250162 -0.199505129
PC5   0.07506535  0.29294217 -0.007136952  0.079061821  0.1250407
2 -0.38689634  0.015250137  0.090082023
PC8  -0.09651306  0.25343194 -0.109466121  0.015521434 -0.0504248
9  0.20564566  0.378135489  0.076663909
PC11 -0.04728893  0.11332517 -0.130755103  0.006006486  0.0480111
4  0.02292825 -0.198032612 -0.300047914
PC12 -0.03569909  0.02938896  0.046577621  0.001638732 -0.3309501
2 -0.06290175  0.087204155 -0.160858368
PC13  0.01158046 -0.02039243 -0.063273909  0.009052647  0.0109953
3  0.01148338 -0.009715174 -0.004611423
              RAD         TAX      PTRATIO            B         LTS
AT
PC1   0.79156616  0.83779482  0.507282755 -0.502407394  0.7667321
84
PC2  -0.32506342 -0.28667263 -0.366218209  0.285602099 -0.0889779
36
PC3   0.32021077  0.24606986 -0.360554349 -0.334580792 -0.2976324
32
PC4  -0.12256314 -0.09569383 -0.261723073 -0.156038625 -0.0642814
59
PC5  -0.18651151 -0.11919955 -0.533592691 -0.315775062  0.3605036
47
PC8  -0.05057452 -0.05209508  0.200064436  0.003098299  0.2670718
72
PC11  0.01576115 -0.04521501  0.075263074  0.008313145  0.1170455
69
PC12  0.04403735  0.08854349 -0.086242461 -0.017167595 -0.0227235
18
PC13 -0.15964602  0.18150635  0.005896553 -0.001124741  0.0061570
38
```

Suppréssion des PC7,PC6,PC10 et PC9 du premier test en fonction de leur pvalue puis regression linéaire

Hide

Hide

```
newbase_4=cbind(housing_PCA_standadize$x,housing["MEDV"])
newbase_4
```

PC1 <dbl>	PC2 <dbl>	PC3 <dbl>	PC4 <dbl>	<
-2.0962230302	0.772348426	0.342603683	0.890892398	0.422652
-1.4558109894	0.591399952	-0.694512011	0.486976617	-0.195682
-2.0725465519	0.599046578	0.166956375	0.738473392	-0.933610
-2.6089217589	-0.006863826	-0.100184990	0.343381425	-1.103863
-2.4557547719	0.097615346	-0.075273718	0.427483833	-1.064870
-2.2126618432	-0.009477633	-0.671716355	0.175736244	-0.626567
-1.3575376559	0.349526292	-0.371631528	0.397740214	1.072086
-0.8412121417	0.577228493	-0.518027787	0.537226588	1.378324
-0.1797503956	0.342179528	-1.348304420	0.245676894	2.347980
-1.0731221380	0.315888821	-0.557917054	0.379556648	1.429116

1-10 of 506 rows | 1-7 of 14… Previous **1** 2 3 4 5 6 … 51 Next

Hide

Hide

```
regression_newbase4=lm(MEDV~PC1+PC2+PC3+PC4+PC5+PC8+PC11+PC12+PC1
3,newbase_4)
regression_newbase4
```

```
Call:
lm(formula = MEDV ~ PC1 + PC2 + PC3 + PC4 + PC5 + PC8 + PC11 + 
    PC12 + PC13, data = newbase_4)

Coefficients:
(Intercept)          PC1          PC2          PC3          PC4  
PC5          PC8         PC11  
     22.533       -2.273        2.195        3.501        1.081  
-2.233       -1.040        1.374  
       PC12         PC13  
      3.084       -3.382  
```

Hide

Hide

```
summary(regression_newbase4)
```

```
Call:
lm(formula = MEDV ~ PC1 + PC2 + PC3 + PC4 + PC5 + PC8 + PC11 + 
    PC12 + PC13, data = newbase_4)

Residuals:
     Min       1Q   Median       3Q      Max 
-16.5289  -2.7838  -0.7749   1.7976  28.9197 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 22.53281    0.21285 105.863  < 2e-16 ***
PC1         -2.27302    0.08608 -26.407  < 2e-16 ***
PC2          2.19491    0.17797  12.333  < 2e-16 ***
PC3          3.50099    0.19113  18.317  < 2e-16 ***
PC4          1.08068    0.23007   4.697 3.42e-06 ***
PC5         -2.23308    0.23319  -9.576  < 2e-16 ***
PC8         -1.04018    0.33853  -3.073  0.00224 ** 
PC11         1.37366    0.49400   2.781  0.00563 ** 
PC12         3.08380    0.51781   5.955 4.91e-09 ***
PC13        -3.38195    0.84544  -4.000 7.29e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.788 on 496 degrees of freedom
Multiple R-squared:  0.7338,	Adjusted R-squared:  0.729 
F-statistic: 151.9 on 9 and 496 DF,  p-value: < 2.2e-16
```

- On selectionne toutes les PCs et on retire celles dont la PV value est supérieure à 1%

Comparons la significativité du test 5 et 4

Hide

Hide

```
anova(regression_newbase4,regression_newbase5)
```

```
Analysis of Variance Table

Model 1: MEDV ~ PC1 + PC2 + PC3 + PC4 + PC5 + PC8 + PC11 + PC12 +
PC13
Model 2: MEDV ~ PC1 + PC2 + PC3 + PC4 + PC5
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1    496 11370
2    500 12944 -4   -1573.6 17.161 3.411e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- Il y a une différence statistique significative entre les deux modèles. On va choisir le modèle 1 qui est plus précis.

On reconstruit la predicted value en revenant sur les variables intiales(Avec le modèle sans les PC dont la pvalue > 1%)

Hide

Hide

```
beta=housing_PCA_standadize$rotation[,c(1,2,3,4,5,8,11,12,13)] %*
% regression_newbase4$coefficients[-1]
beta
```

```
              [,1]
CRIM    -1.0634251
ZN       0.4897006
INDUS    0.3646888
Chase    0.8106361
NOX     -2.4335788
RM       3.1938059
AGE     -0.4097755
DIS     -2.8884653
RAD      2.6018950
TAX     -2.1725453
PTRATIO -2.0873720
B        0.4316314
LTSAT   -3.0796360
```

Conclusion

Le but de ce rapport était de determiner les meilleures variables influant sur le prix du logement dans la banlieue de boston. . Ainsi, notre étude a permis de dégager les résultats suivants :
- Le prix des logements est supérieur dans les zones avec le taux de criminalité le moins élevé .
- Un autre résultat plutôt intéressant concerne le niveau de monoxide d'azote et la distance jusqu'au principaux centres d'emplois.
- D'une part, les gens veulent vivre proches de l'endroit où ils travaillent.
- Cependant, d'autre part, il est raisonnable de suggérer que le niveau de pollution augmente quand on se rapproche de ces grosses zones d'emplois.
- Les coefficients montrent que la distance au travail réduit plus le prix du logement que le niveau NOX. Autrement dit, quand on parle de pollution, les gens sont très sensible à la question. Cependant, ils donnent plus de valeur à un logement proche de leur zone d'emploi et donc avec un certain taux de NOX plutôt qu'un emploi plus loin mais avec un niveau NOX plus faible.
- Depuis, il n'y a aucun doute sur le fait que le niveau de pollution a augmenté et il serait intéressant d'examiner les façons dont cette dernière donnée affecte le prix du logement dans la banlieue de Boston.
- Notons tout de même que les américains sont beaucoup moins sensibles à la notion de pollution que la notre en Europe. Mais que cependant, ils deviennent de plus en plus sensibles car le sujet fait débat depuis de nombreuses années, exemple : COP21