Projet de Modélisation Avancé

Emilian Loric & Alexandre Révillon IS4

2023-03-13

Description multivariée des caractéristiques des maisons.

#Quelles variables expliquent le mieux le prix ?

Premières lignes du jeu de données

```
## Warning: remplacement de l'importation précédente 'ellipsis::check_dots_unnamed'
## par 'rlang::check_dots_unnamed' lors du chargement de 'hms'
```

```
## Warning: remplacement de l'importation précédente 'ellipsis::check_dots_used'
## par 'rlang::check_dots_used' lors du chargement de 'hms'
```

```
## Warning: remplacement de l'importation précédente 'ellipsis::check_dots_empty'
## par 'rlang::check_dots_empty' lors du chargement de 'hms'
```

```
## Warning: 59 parsing failures.
## row col   expected actual          file
##   2 AGE an integer      * 'maisons.txt'
##   9 AGE an integer      * 'maisons.txt'
##   9 TAX an integer      * 'maisons.txt'
##  16 AGE an integer      * 'maisons.txt'
##  20 AGE an integer      * 'maisons.txt'
## ... ... .......... ...... ..............
## See problems(...) for more details.
```

```
## # A tibble: 117 x 8
##    PRICE  SQFT   AGE FEATS NE    CUST  COR     TAX
##    <dbl> <dbl> <int> <dbl> <lgl> <lgl> <lgl> <int>
##  1  2050  2650    13     7 TRUE  TRUE  FALSE  1639
##  2  2080  2600    NA     4 TRUE  TRUE  FALSE  1088
##  3  2150  2664     6     5 TRUE  TRUE  FALSE  1193
##  4  2150  2921     3     6 TRUE  TRUE  FALSE  1635
##  5  1999  2580     4     4 TRUE  TRUE  FALSE  1732
##  6  1900  2580     4     4 TRUE  FALSE FALSE  1534
##  7  1800  2774     2     4 TRUE  FALSE FALSE  1765
##  8  1560  1920     1     5 TRUE  TRUE  FALSE  1161
##  9  1450  2150    NA     4 TRUE  FALSE FALSE    NA
## 10  1449  1710     1     3 TRUE  TRUE  FALSE  1010
## # ... with 107 more rows
```

Description de la signification de chaque variable

```
## spc_tbl_ [117 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ PRICE: num [1:117] 2050 2080 2150 2150 1999 ...
```

```
##  $ SQFT : num [1:117] 2650 2600 2664 2921 2580 ...
##  $ AGE  : int [1:117] 13 NA 6 3 4 4 2 1 NA 1 ...
##  $ FEATS: num [1:117] 7 4 5 6 4 4 4 5 4 3 ...
##  $ NE   : logi [1:117] TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ CUST : logi [1:117] TRUE TRUE TRUE TRUE TRUE FALSE ...
##  $ COR  : logi [1:117] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ TAX  : int [1:117] 1639 1088 1193 1635 1732 1534 1765 1161 NA 1010 ...
##  - attr(*, "problems")= tibble [59 x 5] (S3: tbl_df/tbl/data.frame)
##   ..$ row     : int [1:59] 2 9 9 16 20 22 23 26 29 29 ...
##   ..$ col     : chr [1:59] "AGE" "AGE" "TAX" "AGE" ...
##   ..$ expected: chr [1:59] "an integer" "an integer" "an integer" "an integer" ...
##   ..$ actual  : chr [1:59] "*" "*" "*" "*" ...
##   ..$ file    : chr [1:59] "'maisons.txt'" "'maisons.txt'" "'maisons.txt'" "'maisons.txt'" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   PRICE = col_double(),
##   ..   SQFT = col_double(),
##   ..   AGE = col_integer(),
##   ..   FEATS = col_double(),
##   ..   NE = col_logical(),
##   ..   CUST = col_logical(),
##   ..   COR = col_logical(),
##   ..   TAX = col_integer()
##   .. )
```

PRIX = Prix de vente (en centaines de dollars)

SQFT = Surface habitable en pieds carrés

AGE = Âge de la maison (années)

CARACTÉRISTIQUES = Nombre de 11 caractéristiques (lave-vaisselle, réfrigérateur, micro-ondes, broyeur, laveuse, interphone, lucarne(s), compacteur, sèche-linge, etc, broyeur, laveuse, interphone, puits de lumière, compacteur, sécheuse, aménagement pour les handicapés, accès à la télévision par câble) câble)

NE = Situé dans le secteur nord-est de la ville (1) ou non (0)

CUST = Construit sur mesure (1) ou non (0)

COR = Emplacement d'angle (1) ou non (0)

TAX = Taxes annuelles ($)

Nombre d'observations et nombre de caractéristiques

```
## [1] 117   8
```

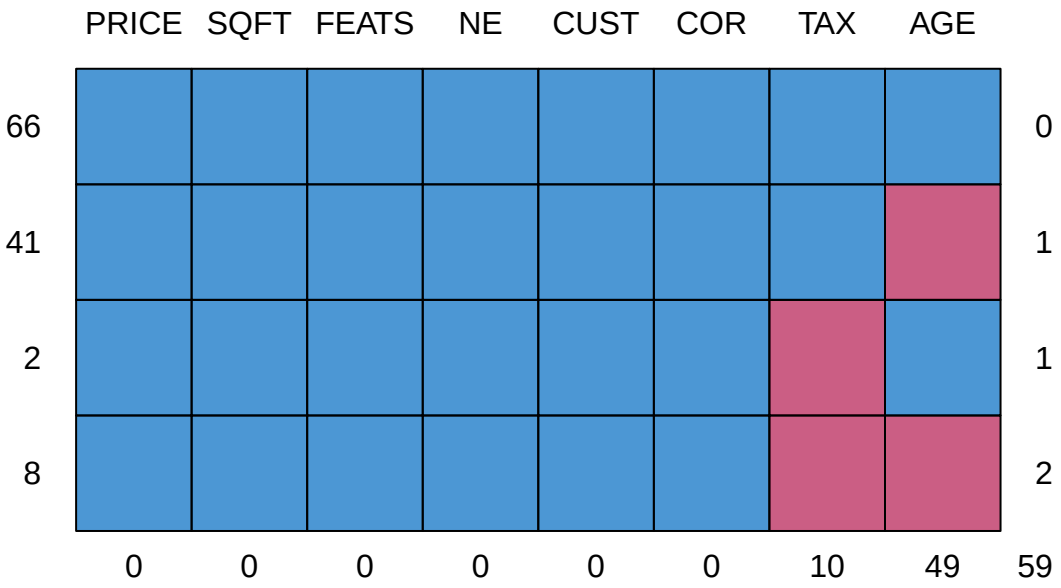
On a dans le jeu de données 117 observations pour 8 variables.

Desciption univariée de chaque variable

```
##      PRICE           SQFT           AGE            FEATS          NE
##  Min.   : 540   Min.   : 837   Min.   : 1.00   Min.   :0.00   Mode :logical
##  1st Qu.: 780   1st Qu.:1280   1st Qu.: 5.75   1st Qu.:3.00   FALSE:39
##  Median : 960   Median :1549   Median :13.00   Median :4.00   TRUE :78
##  Mean   :1063   Mean   :1654   Mean   :14.97   Mean   :3.53
##  3rd Qu.:1200   3rd Qu.:1894   3rd Qu.:19.25   3rd Qu.:4.00
##  Max.   :2150   Max.   :3750   Max.   :53.00   Max.   :8.00
##                                NA's   :49
```

```
##     CUST            COR               TAX
##  Mode :logical   Mode :logical   Min.   : 223.0
##  FALSE:90        FALSE:95        1st Qu.: 600.0
##  TRUE :27        TRUE :22        Median : 731.0
##                                  Mean   : 793.5
##                                  3rd Qu.: 919.0
##                                  Max.   :1765.0
##                                  NA's   :10
```

Il y a 2 variables comportant des données manquantes, à savoir AGE et TAX. Pour la variable AGE, il y a 41.88% de données manquantes et 8.54% pour la variable TAX.




```
##    PRICE SQFT FEATS NE CUST COR TAX AGE
## 66     1    1     1  1    1   1   1   1  0
## 41     1    1     1  1    1   1   1   0  1
## 2      1    1     1  1    1   1   0   1  1
## 8      1    1     1  1    1   1   0   0  2
##        0    0     0  0    0   0  10  49 59
```

MCAR, MAR ou MNAR

On visualise entre différentes variables pour essayer détecter visuellement d'éventuels MCAR, MAR ou MNAR.

```
## Le chargement a nécessité le package : colorspace
```

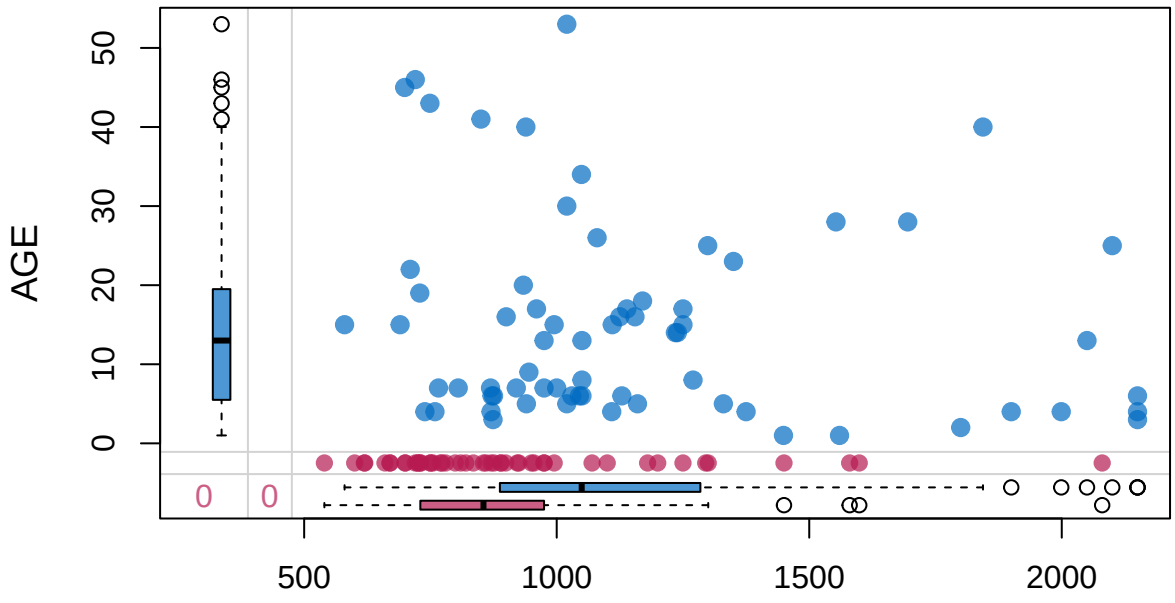
```
## Le chargement a nécessité le package : grid
```

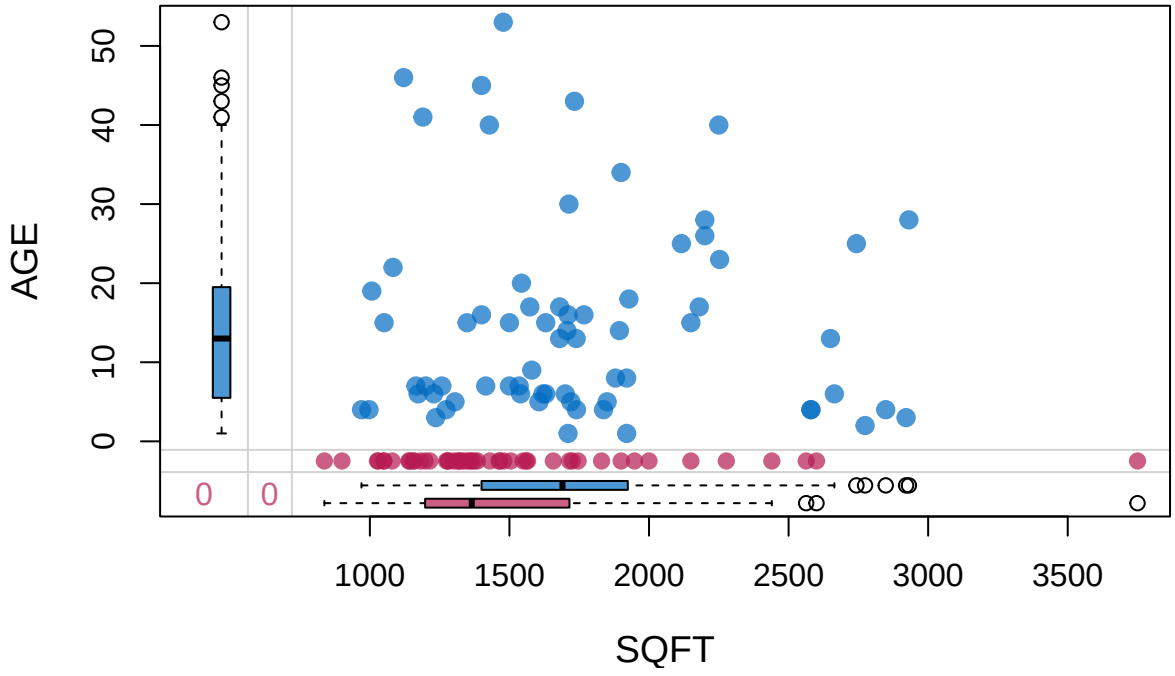
```
## VIM is ready to use.
```

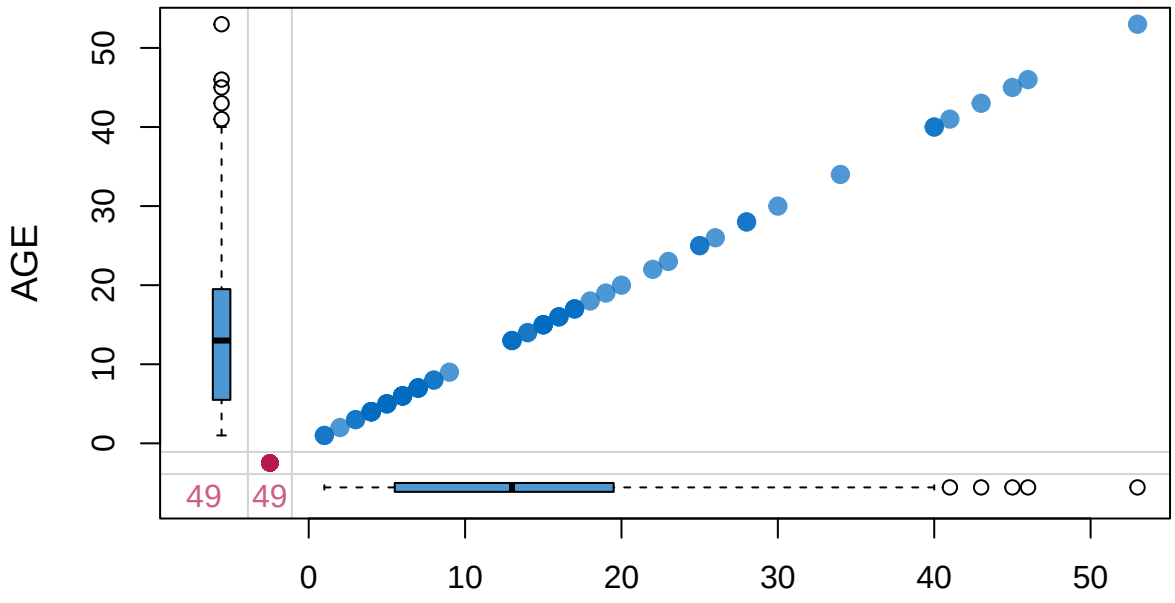
```
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
```

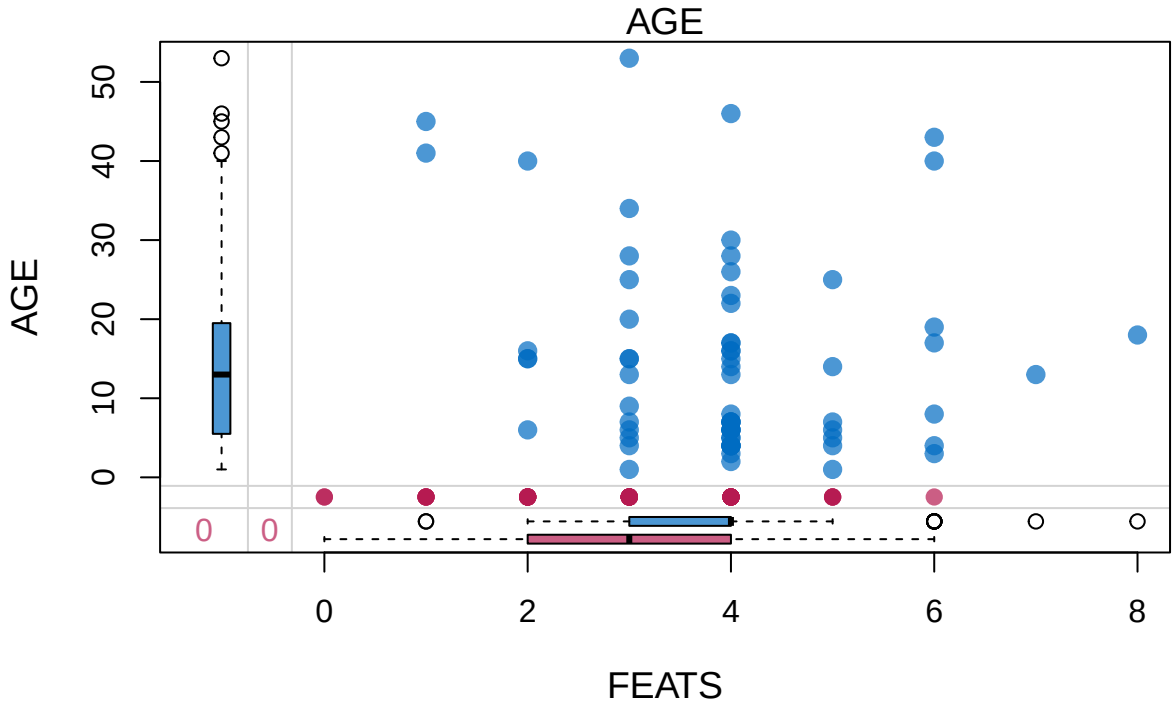
```
##
## Attachement du package : 'VIM'
```

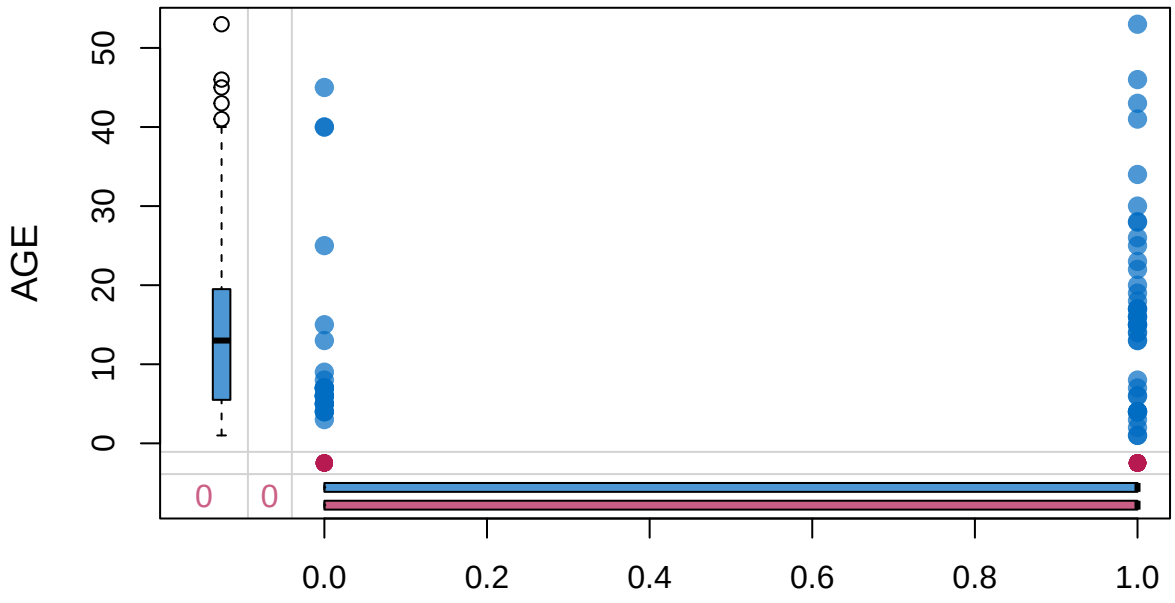
```
## L'objet suivant est masqué depuis 'package:datasets':
##
##     sleep
```

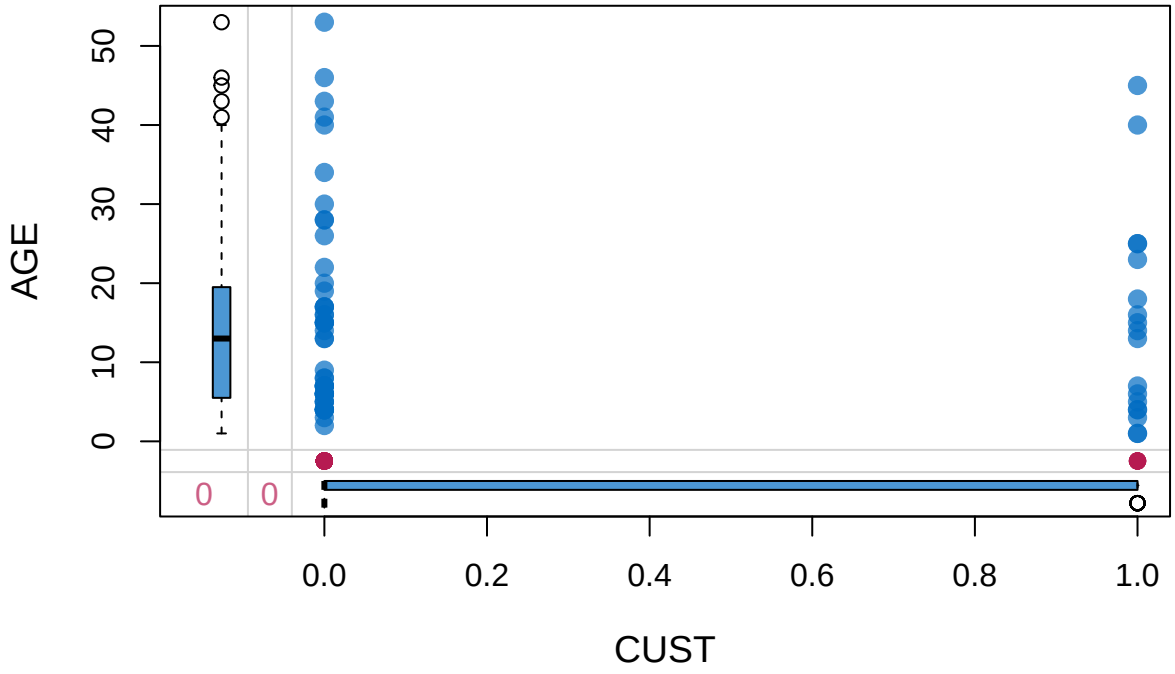

AGE
0
10
20
30
40
50
0
0
500
1000
1500
2000
PRICE

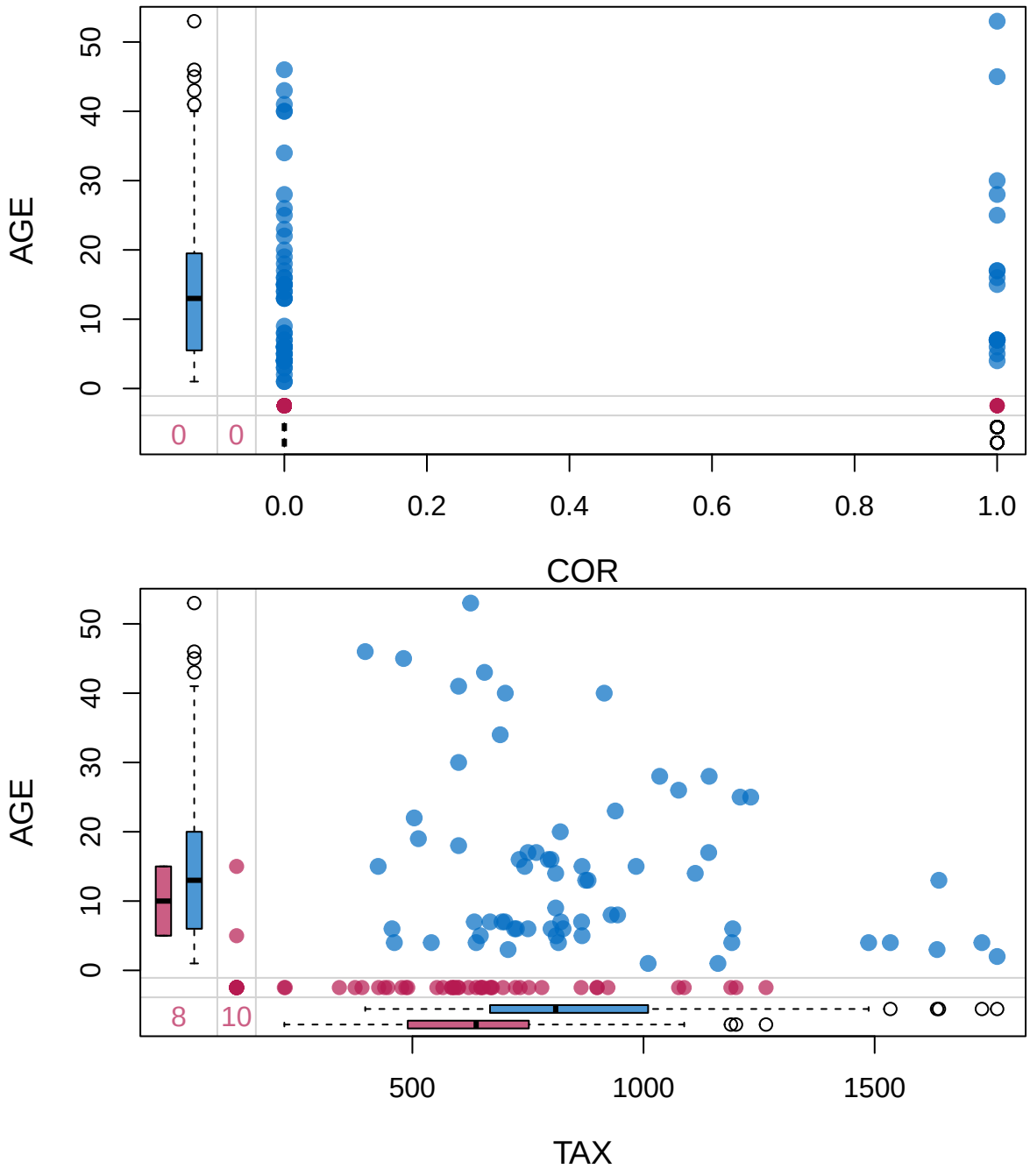

AGE
0
10
20
30
40
50
0
0
1000
1500
2000
2500
3000
3500
SQFT


AGE
0
10
20
30
40
50
49
49
0
10
20
30
40
50
AGE


AGE
0
10
20
30
40
50
0
0
0
2
4
6
8
FEATS


AGE
0
10
20
30
40
50
0
0
0.0
0.2
0.4
0.6
0.8
1.0
NE


AGE
0
10
20
30
40
50
0
0
0.0
0.2
0.4
0.6
0.8
1.0
CUST



Test de Student sur la variable PRICE entre le groupe avec données manquantes sur AGE et le groupe sans donnée manquante sur AGE

```
##
##  Welch Two Sample t-test
##
## data:  maisons$PRICE[is.na(maisons$AGE)] and maisons$PRICE[!is.na(maisons$AGE)]
## t = -3.732, df = 114.92, p-value = 0.0002969
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -369.9171 -113.3939
```

```
## sample estimates:
## mean of x mean of y
##  922.2857 1163.9412
```

La distribution de la variable PRIX est significativement différente entre les observations qui ont pour donnée manquante AGE et les autres. Les maisons pours lesquels nous n'avons pas la variable AGE sont ceux qui ont un prix en moyenne moins élevé que ceux dont on a l'AGE. Par conséquent, la répartition des données manquantes ne seraient donc pas complètement aléatoire.

Test de Student sur la variable FEATS entre le groupe avec données manquantes sur AGE et le groupe sans donnée manquante sur AGE

```
##
##  Welch Two Sample t-test
##
## data:  maisons$FEATS[is.na(maisons$AGE)] and maisons$FEATS[!is.na(maisons$AGE)]
## t = -4.2605, df = 101.51, p-value = 4.57e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.5421200 -0.5623218
## sample estimates:
## mean of x mean of y
##  2.918367  3.970588
```

La pvalue est aussi très inférieur au seuil d'erreur de 5%, on rejette donc l'hypothèse que les données manquante sur la variable AGE soient complètement aléatoires.

Test de Student sur la variable SQFT entre le groupe avec données manquantes sur AGE et le groupe sans donnée manquante sur AGE

```
##
##  Welch Two Sample t-test
##
## data:  maisons$SQFT[is.na(maisons$AGE)] and maisons$SQFT[!is.na(maisons$AGE)]
## t = -2.2786, df = 101.5, p-value = 0.02479
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -413.06699  -28.58908
## sample estimates:
## mean of x mean of y
##  1525.510  1746.338
```

Test de Student sur la variable TAX entre le groupe avec données manquantes sur AGE et le groupe sans donnée manquante sur AGE

```
##
##  Welch Two Sample t-test
##
## data:  maisons$TAX[is.na(maisons$AGE)] and maisons$TAX[!is.na(maisons$AGE)]
## t = -3.7567, df = 99.148, p-value = 0.0002908
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -315.90550  -97.53722
## sample estimates:
## mean of x mean of y
##  665.9756  872.6970
```

Graphique des corrélations entre chaque variable

```
## corrplot 0.92 loaded
```

```
##        PRICE   SQFT    AGE  FEATS     NE   CUST    COR    TAX
## PRICE 100.00  84.48 -16.87  42.03  16.78  55.53  -7.93  87.57
## SQFT   84.48 100.00  -3.97  39.49  14.50  52.01   4.05  85.86
## AGE   -16.87  -3.97 100.00 -18.78  22.68  -1.18  13.64 -29.18
## FEATS  42.03  39.49 -18.78 100.00  19.00  24.20  -4.15  44.17
## NE     16.78  14.50  22.68  19.00 100.00   4.30  -7.73  19.74
## CUST   55.53  52.01  -1.18  24.20   4.30 100.00  -0.40  46.99
## COR    -7.93   4.05  13.64  -4.15  -7.73  -0.40 100.00  -6.00
## TAX    87.57  85.86 -29.18  44.17  19.74  46.99  -6.00 100.00
```

Etant données qu l'on observe des corrélations entre la variable AGE et d'autres variables telles que PRICE ou encore TAX, on peut en déduire que les valeurs manqantes ne sont pas MCAR comme nous l'avions aussi conclut lors des test de Student. La présence des valeurs manquantes ne pouvant être expliqué que par la variable AGE elle-même, elle ne sont pas MNAR non plus. On en déduit donc que les valeurs manquantes pour la variable AGE sont MAR.

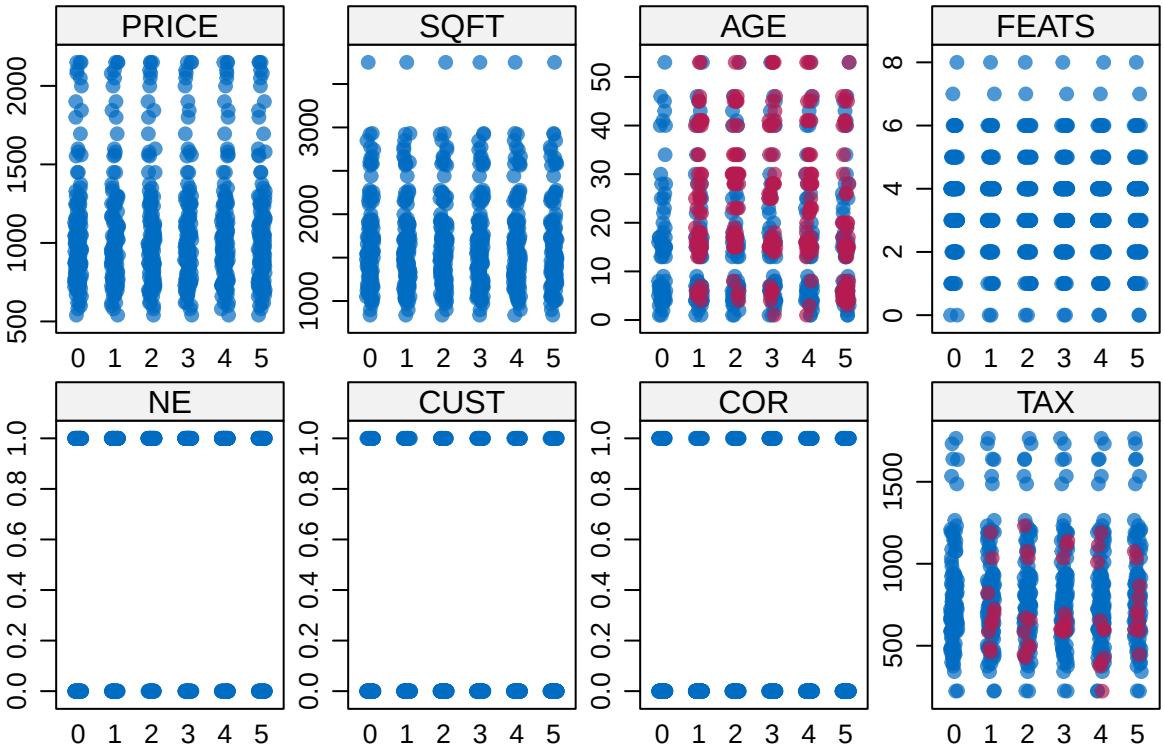
Dans ce cas, une approche pertinente d'imputation des données est de réaliser une imputation multiple avec des valeurs plausibles.

#MICE

Jeu de données n°1 de l'imputation multiple MICE

```
##      PRICE           SQFT           AGE            FEATS          NE
##  Min.   : 540   Min.   : 837   Min.   : 1.00   Min.   :0.00   Mode :logical
##  1st Qu.: 780   1st Qu.:1280   1st Qu.: 7.00   1st Qu.:3.00   FALSE:39
##  Median : 960   Median :1549   Median :16.00   Median :4.00   TRUE :78
##  Mean   :1063   Mean   :1654   Mean   :19.26   Mean   :3.53
##  3rd Qu.:1200   3rd Qu.:1894   3rd Qu.:28.00   3rd Qu.:4.00
##  Max.   :2150   Max.   :3750   Max.   :53.00   Max.   :8.00
##     CUST            COR               TAX
##  Mode :logical   Mode :logical   Min.   : 223.0
##  FALSE:90        FALSE:95        1st Qu.: 600.0
##  TRUE :27        TRUE :22        Median : 725.0
##                                  Mean   : 787.9
##                                  3rd Qu.: 915.0
##                                  Max.   :1765.0
```

Graphique sur la distribution des données manquante des 5 jeux de données issus de MICE

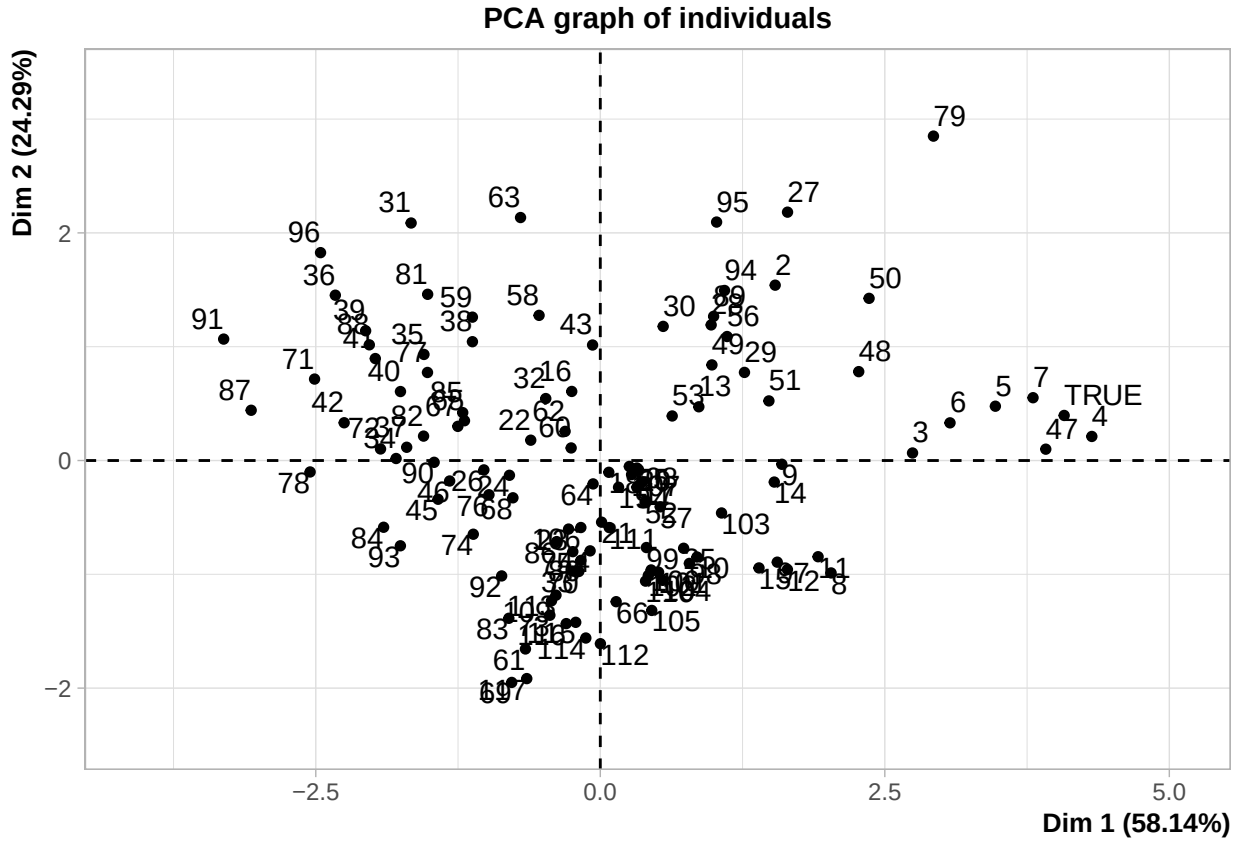


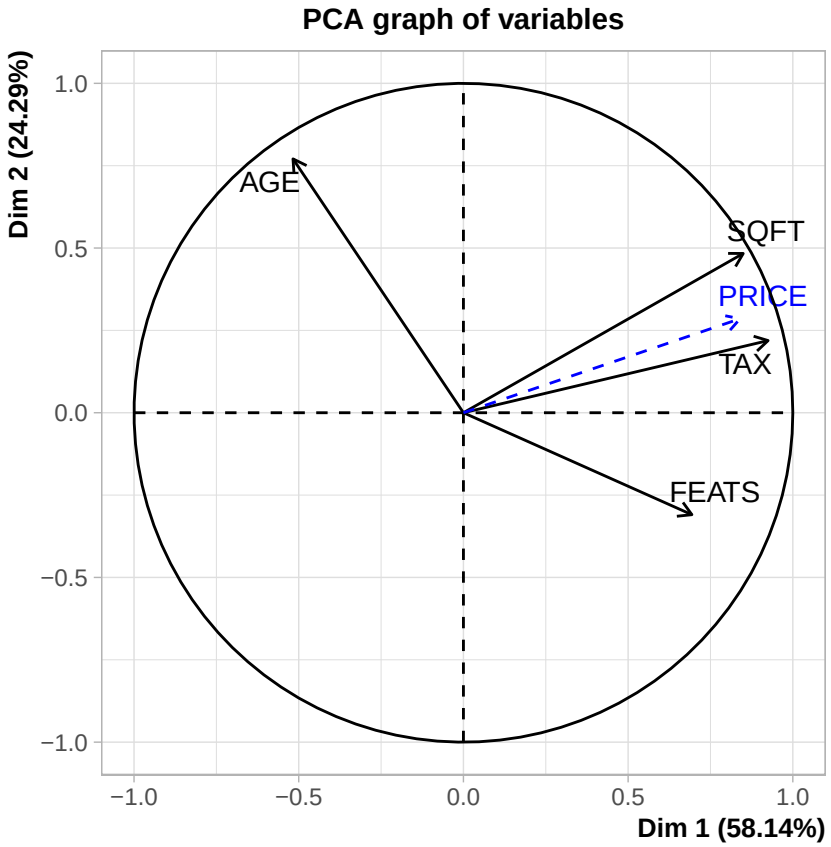
La distribution des données manquantes semble cohérente pour chaque jeu de données.

Fusion des 5 jeux de données imputés

Pour le reste de l'étude, nous avons besoin d'avoir un jeux de données sasn données manquante. Pour cela, nous allons fusionner les différents jeux de données obtenus par les différentes imputations en faisant la moyenne pour chaque individu des données provenant des 5 jeux de données.

Etude de la multicolinéarité






```
##        eigenvalue percentage of variance cumulative percentage of variance
## comp 1 2.32570713              58.142678                           58.14268
## comp 2 0.97141756              24.285439                           82.42812
## comp 3 0.61088243              15.272061                           97.70018
## comp 4 0.09199288               2.299822                          100.00000
```

La dernière composante représente 2.29% de l'information, la question se pose de savoir si nous devons la considérer comme négligeable ou non, et donc s'alerter d'une éventuelle multicolinéarité.

Pour étudier les possibles multicolinéarités, il faut d'abord réaliser un modèle de régression prennant en compte toutes les variables. Ici comme nous avons 5 jeux de données à cause de l'imputation multiple, nous avons donc 5 modèles différents.

Le critère pour étudier la multicolinéarité que nous allons étudier est le critère VIF:

$VIF(X_i) = \frac{1}{1-R_i^2}$

Lorsque ce critère VIF est élevé, c'est un signe de multicolinéarité évidente.

```
## Le chargement a nécessité le package : carData
```

```
##     SQFT      AGE    FEATS       NE     CUST      COR      TAX
## 5.877243 2.055413 1.463502 1.320936 1.397624 1.026503 6.695619
```

On remarque que le VIF des variables TAX et SQFT sont suffisemment élevé pour conclure d'une multicolinéartié entre ces 2 variables.

Pour la suite de l'étude, nous allons donc utiliser des méthode permettant de selectionner les variables interessantes pour réaliser un modèle de regression linéaire. Nous utiliserons les méthodes Lasso, PCR, PLS, Ridge, Elastic-net, et stepAIC pour lesquelles nous étudierons leurs performances afin de garder le meilleur modèle. Plus précisemment, nous utiliserons le critère de RMSEP.

#Selection des variables (Ridge, Lasso, PCR, PLS, Elastic-net et stepAIC)

stepAIC

```
##
## Call:
## lm(formula = PRICE ~ SQFT + CUST + COR + TAX, data = dm_moy_complete)
##
## Coefficients:
## (Intercept)         SQFT     CUSTTRUE      CORTRUE          TAX
##    143.2124       0.2093     128.2496     -70.9646       0.7076
```

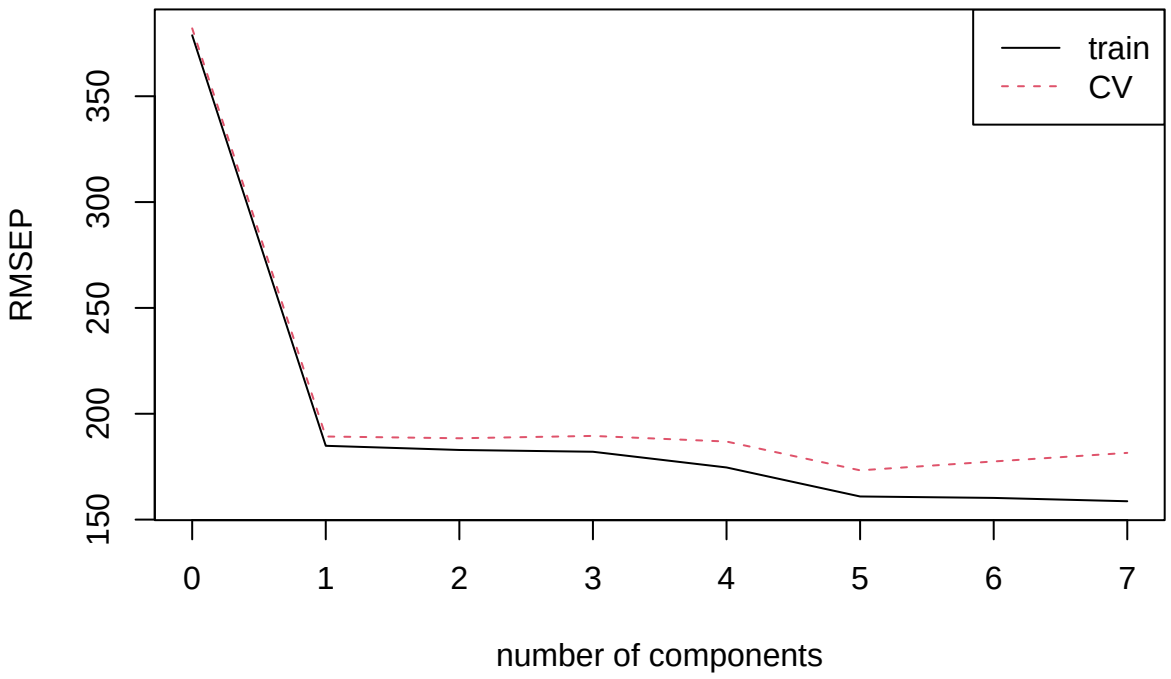
Suite au step AIC, les variables qui expliquent le mieux le “PRICE” sont “CUST”, “TAX”, “SQFT” et “COR”.

Maintenant on va calculer son RMSEP afin de pouvoir comparer ce modèle avec les autres que nous allons faire par la suite.

```
## [1] 175.1206
```

PCR

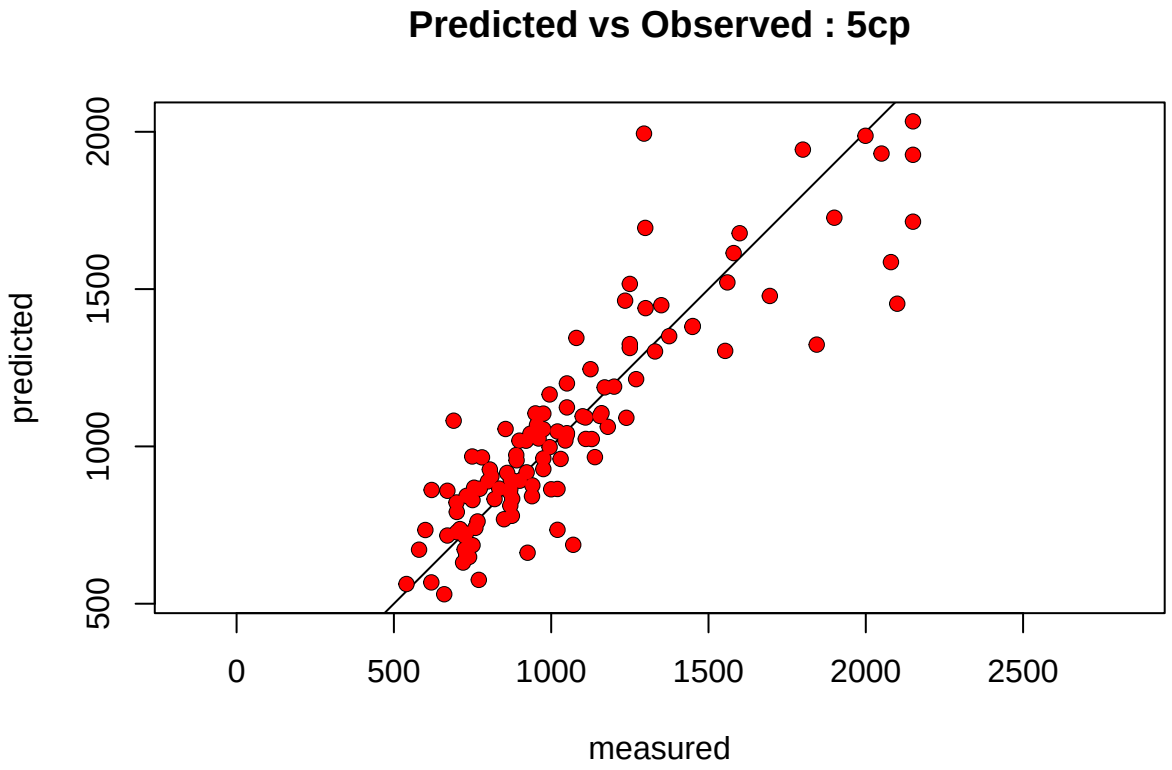
```
##
## Attachement du package : 'pls'

## L'objet suivant est masqué depuis 'package:corrplot':
##
##     corrplot

## L'objet suivant est masqué depuis 'package:stats':
##
##     loadings

## Principal component regression , fitted with the singular value decomposition algorithm.
## Cross-validated using 117 leave-one-out segments.
## Call:
## pcr(formula = PRICE ~ ., data = dm_moy_complete, scale = TRUE,     validation = "LOO", jackknife = T
```



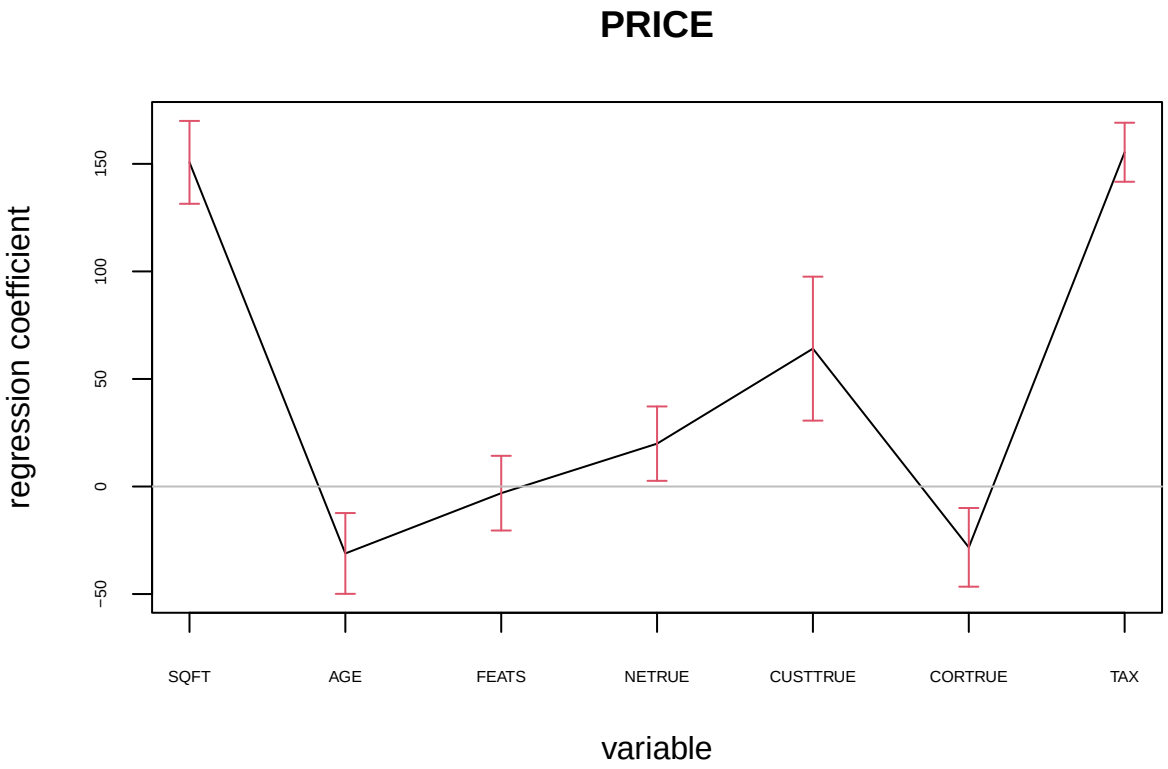

```
## Data:    X dimension: 117 7
##  Y dimension: 117 1
## Fit method: svdpc
## Number of components considered: 7
##
## VALIDATION: RMSEP
## Cross-validated using 117 leave-one-out segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           382.1    189.3    188.4    189.5    186.9    173.3    177.5
## adjCV        382.1    189.2    188.4    189.5    186.8    173.2    177.4
##        7 comps
## CV       181.5
## adjCV    181.4
##
## TRAINING: % variance explained
##        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X        37.99    55.78    71.01    83.86    92.32    98.88   100.00
## PRICE    76.18    76.69    76.90    78.75    81.95    82.11    82.45
```

A l'aide du graphique et du tableau des valeurs de RMSEP, on cherche le nombre de composante qui minimise le RMSEP. Ici on va donc choisir 5 composantes pour une valeur de RMSEP de 173.3. On a ici un RMSEP inférieur à celui obtenu à l'aide de stepAIC. Le modèle sera donc meilleur.

Voici les prédictions graphiquement (obtenues par validation croisée) :



Les coefficients et leur significativité est obtenue avec :




```
## Response PRICE (5 comps):
##          Estimate Std. Error  Df t value  Pr(>|t|)
## SQFT     150.6990    19.2619 116  7.8237 2.625e-12 ***
## AGE      -31.1248    18.7798 116 -1.6574   0.10015
```

```
## FEATS     -3.0967    17.3611 116 -0.1784   0.85874
## NETRUE    19.9254    17.2874 116  1.1526   0.25144
## CUSTTRUE  64.0958    33.4594 116  1.9156   0.05787 .
## CORTRUE  -28.2824    18.2863 116 -1.5466   0.12467
## TAX      155.3987    13.7195 116 11.3268 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pour un seuil d'erreur de $\alpha = 5\%$, on observe que seul les variables SQFT et TAX sont significative.

Pour obtenir en plus la valeur de l'intercept, nous devons faire la manipulation suivante:

```
## , , 5 comps
##
##                   PRICE
## (Intercept) 190.284518
## SQFT        150.698980
## AGE         -31.124817
## FEATS        -3.096718
## NETRUE       19.925404
## CUSTTRUE     64.095810
## CORTRUE     -28.282417
## TAX         155.398706
```

Les variables qui ont le plus d'importance dans le modèle obtenu sont les variables "SQFT", "TAX" et "CUST". Nous pouvons noter que ces variables sont égalmenet dans le modèle issu de stepAIC.

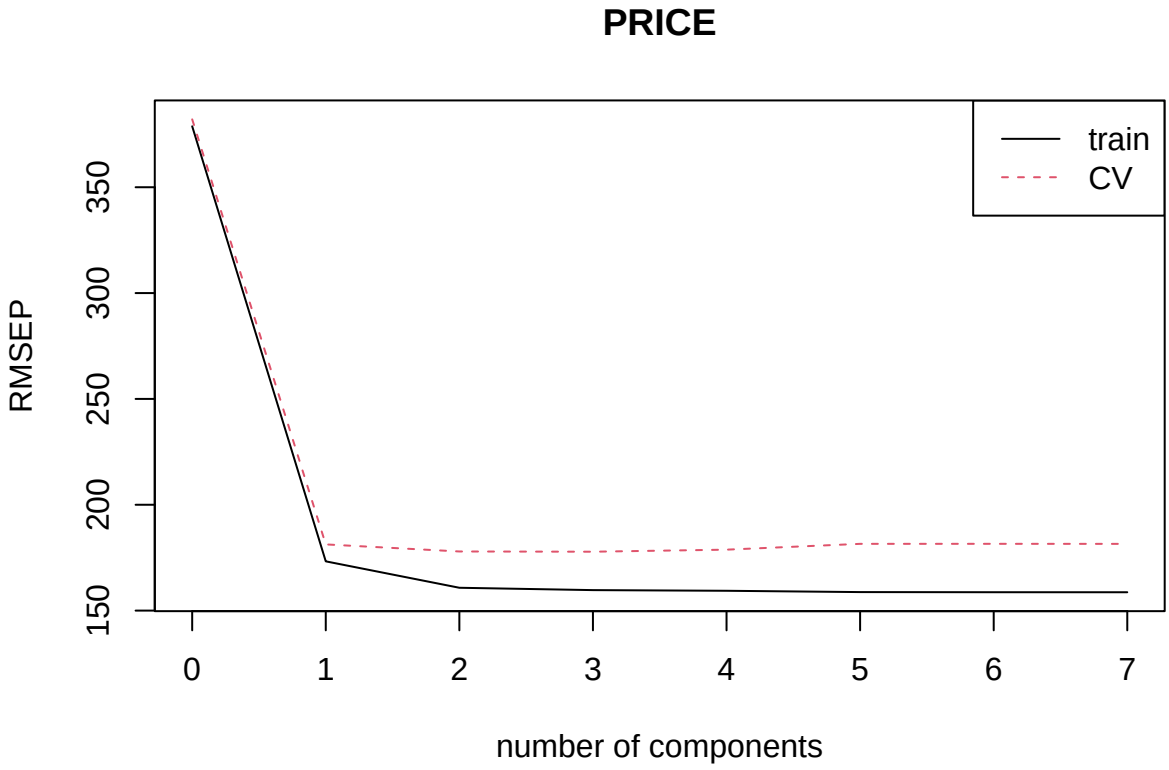
```
##        SQFT         AGE       FEATS          NE        CUST         COR
##   0.2877457  -2.6194838  -2.2033078  42.0871442 151.4776043 -72.0717888
##         TAX
##   0.5136442
```

Le modèle de régression obtenu est donc :

$PRICE = 190.2845 + 0.2877 * SQFT - 2.6194 * AGE - 2.2033 * FEATS + 42.0871 * NE + 151.4776 * CUST - 72.0717 * COR + 0.5136 * TAX + erreur$

Ce modèle à un RMSEP de 173.3, ce qui est un meilleur score de prediction que le modèle obtenu à l'aide de stepAIC.

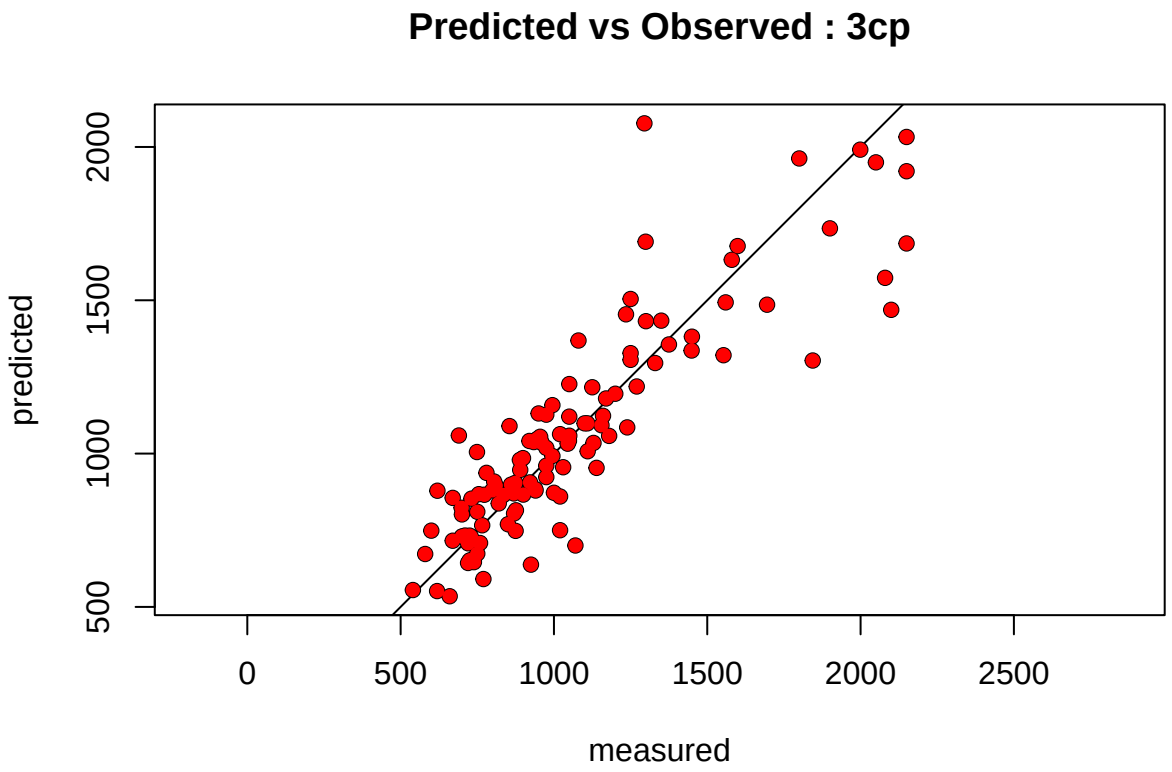
##PLS



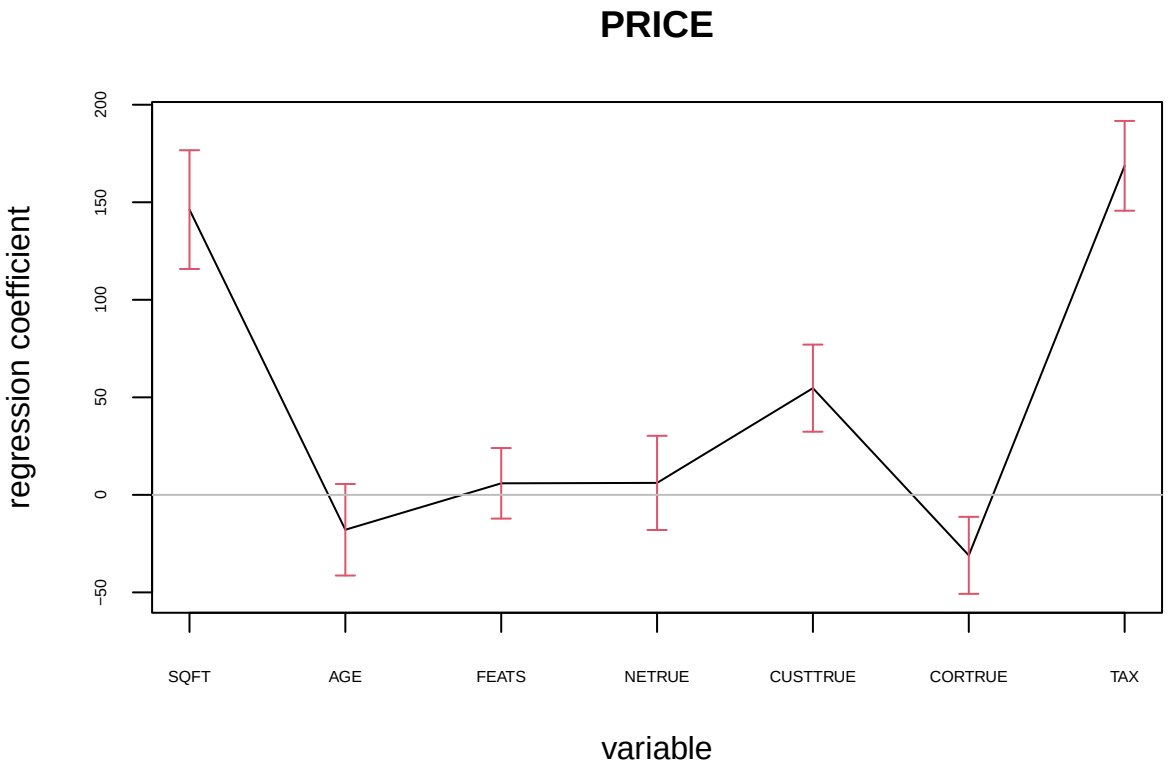

```
## Data:    X dimension: 117 7
##  Y dimension: 117 1
## Fit method: kernelpls
## Number of components considered: 7
##
## VALIDATION: RMSEP
## Cross-validated using 117 leave-one-out segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           382.1    181.3    177.9    177.8    178.8    181.5    181.6
## adjCV        382.1    181.2    177.9    177.8    178.7    181.4    181.4
##        7 comps
## CV       181.5
## adjCV    181.4
##
## TRAINING: % variance explained
##        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X        37.82    49.73    62.81    75.14    79.18    90.92   100.00
## PRICE    79.08    81.98    82.23    82.30    82.44    82.45    82.45
```

A l'aide du graphique et du tableau des valeurs de RMSEP, on cherche le nombre de composante qui minimise le RMSEP. Ici on va donc choisir 3 composantes pour une valeur de RMSEP de 177.8. On a ici un RMSEP inférieur à ceux obtenus précedemment.

Voici les prédictions graphiquement (obtenues par validation croisée) :



Les coefficients et leur significativité est obtenue avec :




```
## Response PRICE (3 comps):
##          Estimate Std. Error  Df t value  Pr(>|t|)
## SQFT     146.2336    30.4360 116  4.8046 4.680e-06 ***
## AGE      -17.8898    23.4639 116 -0.7624   0.44734
```

```
## FEATS      5.9208    18.0848 116  0.3274   0.74396
## NETRUE     6.1326    24.1413 116  0.2540   0.79992
## CUSTTRUE  54.7001    22.3278 116  2.4499   0.01578 *
## CORTRUE  -31.0043    19.7423 116 -1.5704   0.11903
## TAX      168.6777    23.0171 116  7.3284 3.367e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pour un seuil d'erreur de $\alpha = 5\%$, on observe que seul les variables SQFT, TAX et CUSTTRUE sont significative.

Pour obtenir en plus la valeur de l'intercept, nous devons faire la manipulation suivante:

```
## , , 3 comps
##
##                   PRICE
## (Intercept) 151.959694
## SQFT        146.233632
## AGE         -17.889839
## FEATS         5.920810
## NETRUE        6.132615
## CUSTTRUE     54.700128
## CORTRUE     -31.004312
## TAX         168.677742
```

Les variables qui ont le plus d'importance dans le modèle obtenu sont les variables "SQFT", "TAX" et "CUST". Nous pouvons noter que ces variables sont égalmenet dans le modèle issu de stepAIC.

```
##        SQFT         AGE       FEATS          NE        CUST         COR
##   0.2792195  -1.5056199   4.2126430  12.9535269 129.2727920 -79.0079667
##         TAX
##   0.5575358
```

Le modèle de régression obtenu est donc :

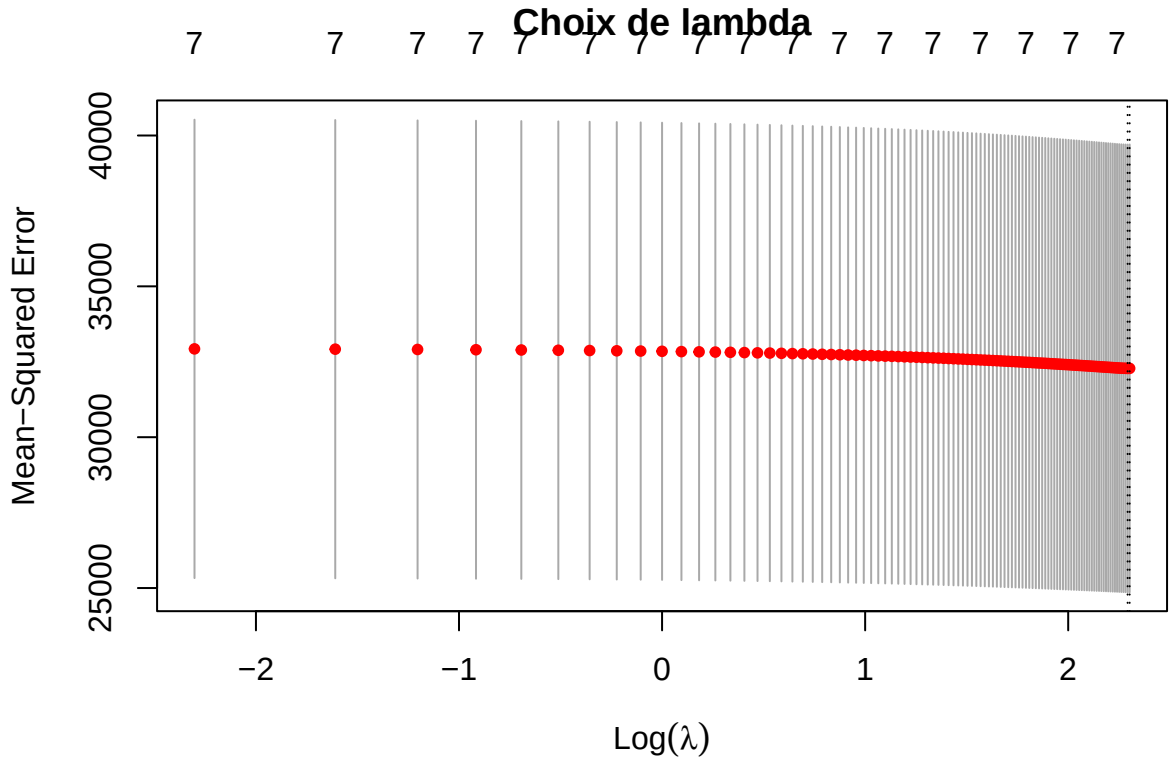
$PRICE = 151.9596 + 0.2792 * SQFT - 1.5056 * AGE - 4.2126 * FEATS + 12.9535 * NE + 129.2727 * CUST - 79.0079 * COR + 0.5575 * TAX + erreur$

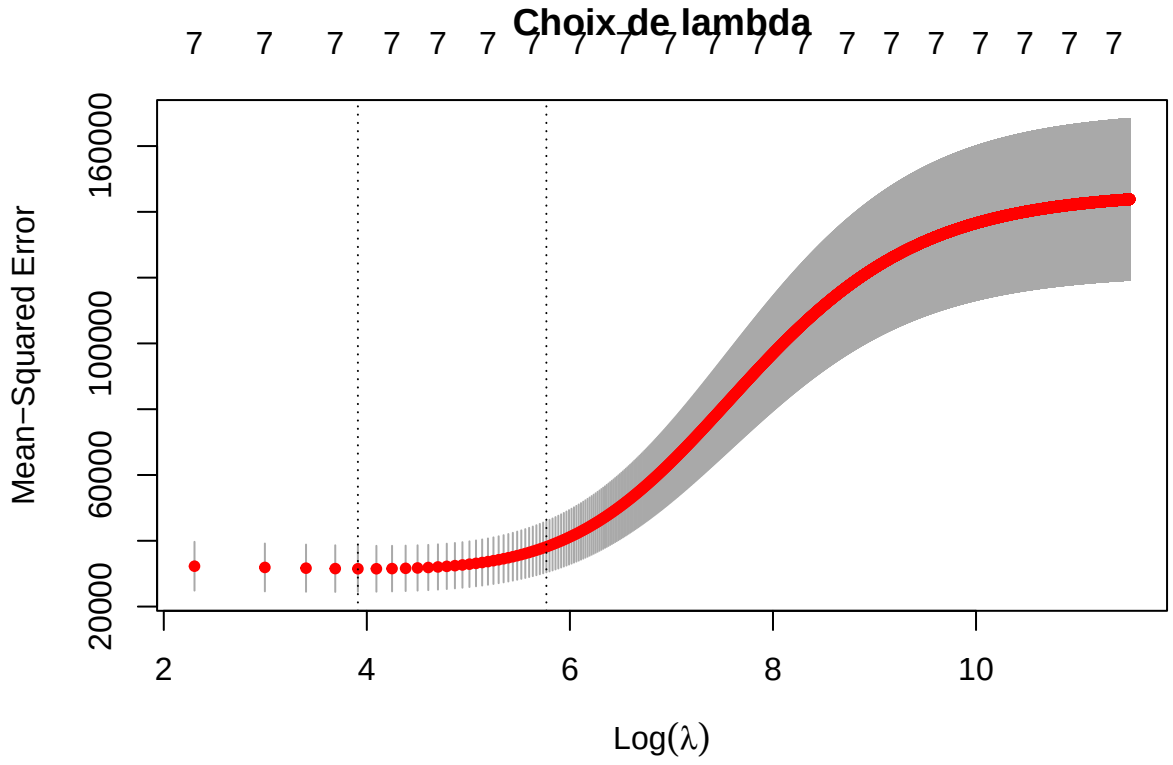
Ce modèle à un RMSEP de 177.8, ce qui est le pire score de prediction des modèles obtenus pour le moment.

Ridge

```
## Le chargement a nécessité le package : Matrix
```

```
## Loaded glmnet 4.1-6
```






```
## [1] 50
## [1] 31494.72
## [1] 177.4675
```

Coefficients sur les données normaliséss

Les coefficients du modèle Ridge optimal sur les données normalisées sont :

```
##       SQFT        AGE      FEATS         NE       CUST        COR        TAX
## 125.880675 -12.935194  16.404309   8.656865  57.139228 -25.001702 163.614735
```

Les variables les plus influentes issues du modèle de régression de Ridge sont “TAX”,“SQFT” et “CUST”.

Coefficients sur les données d’origine

Les coefficients du modèle Ridge optimal sur les données d’origines sont :

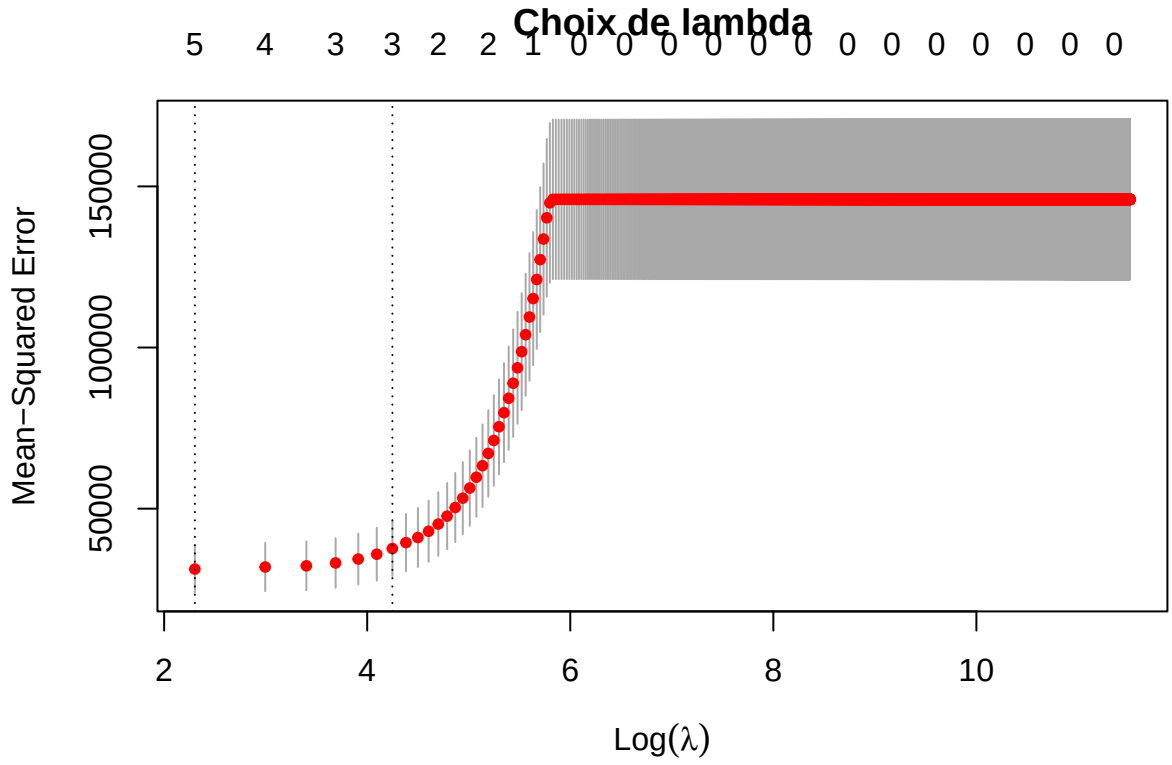
```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 187.4300535
## SQFT          0.2403574
## AGE          -1.0886339
## FEATS        11.6716286
## NETRUE       18.2853367
## CUSTTRUE    135.0371178
## CORTRUE     -63.7115783
## TAX           0.5408008
```

Remarque : Nous n’avons pas leur significativité avec glmnet.

Lasso

C’est la même chose que pour Ridge, sauf que le paramètre α vaut 1.

Graphique pour déterminer le lambda optimal



###

Lambda optimal

```
## [1] 10
```

```
## [1] 31255.63
```

RMSEP modèle de Lasso

```
## [1] 176.7926
```

Coefficients sur les données normalisées

Les coefficients du modèle Lasso optimal sur les données normalisées sont :

```
##       SQFT        AGE      FEATS         NE       CUST        COR        TAX
## 104.139039   0.000000   6.258628   0.000000  47.823635 -17.495805 209.383918
```

Les variables les plus influentes issues du modèle de régression de Lasso sont “TAX”,“SQFT” et “CUST”.

Coefficients sur les données d’origine

Les coefficients du modèle Lasso optimal sur les données normalisées sont :

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 155.5352521
## SQFT          0.1988438
## AGE           .
## FEATS         4.4529999
## NETRUE        .
## CUSTTRUE    113.0215792
## CORTRUE     -44.5843787
## TAX           0.6920831
```

Elastic-net

La régression de type elastic-net consiste à combiner Ridge L1 et Lasso L2 pour améliorer la performance de la prédiction et la stabilité du modèle. Cette méthode est souvent utilisée pour traiter des problèmes ou le nombre de variables explicatives est important et ou il existe des relations de corrélation entre ces variables comme ici.

Par itération, nous déterminons le *alpha* minimisant le RMSEP.

Itération 1

```
##   alpha    rmsep   lambda
## 3   0.3 176.2995 32.39351
## 4   0.4 176.1317 26.66388
## 5   0.5 176.1649 21.33110
```

Itération 2

```
##    alpha    rmsep   lambda
## 12  0.41 176.1256 26.01354
## 13  0.42 176.1251 25.39417
## 14  0.43 176.1267 24.80361
```

Itération 3

```
##    alpha    rmsep   lambda
## 12 0.421 176.1251 25.33385
## 13 0.422 176.1242 25.27382
## 14 0.423 176.1243 25.21407
```

Le RMSEP minimum de elastic-net est 176.1242 pour alpha=0.422 et on obtient un lambda optimal de 25.2738

Coefficients du modèle elastic-net

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 165.9330499
## SQFT          0.2128493
## AGE           .
## FEATS         6.4677079
## NETRUE        .
## CUSTTRUE    113.8366206
## CORTRUE     -42.9106075
## TAX           0.6397881
```

Conclusion

Avec un RMSEP de 173.3 , le modèle obtenu par PCR est le meilleur modèle, c'est-à-dire avec la plus faible erreur moyenne de prédictions parmi les modèles obtenus par step AIC, PCR, PLS, Ridge, Lasso et elastic-net.

Le modèle de PCR : $PRICE = 190.2845 + 0.2877 * SQFT - 2.6194 * AGE - 2.2033 * FEATS + 42.0871 * NE + 151.4776 * CUST - 72.0717 * COR + 0.5136 * TAX + erreur$.

C'est donc ce modèle qui nous donne les meilleurs prédictions selon le critère RMSEP.

Par analyse des coefficents normalisés, les variables qui expliquent le mieux le prix des maisons sont "SQFT", "TAX" et "CUST".

Par ailleurs, les variables SQFT et TAX sont celles qui sont gardés par tous les modèles. La variable CUST est celle qui ensuite revient le plus souvent parmi toutes les autres.

Ainsi, les variables qui expliquent le mieux le prix sont le nombre de mètre carrés (ici les pieds carrés), le montant des taxes et le fait que la maison soit situé sur un coin de rue ou non.

Ce qui quand on y réfléchit est totalement en raccord avec le fait que les prix des maisons sont calculé en fonction du nombre de metre carré.

Pourquoi une maison situé sur un coin de rue vaudrait plus cher ? Les coins de rue sont des intersections pour les autombilites qui y rencontre très souvent un stop. Ainsi, la pollution sonore est plus faibel pour ces maisons, ce qui explique un prix de maison plus important.

Annexe : notre code R

```
library(readr)
maisons <- read_table2("maisons.txt", col_types = cols(AGE = col_integer(),
    NE = col_logical(), CUST = col_logical(),
    COR = col_logical(), TAX = col_integer()),
    skip = 17)
```

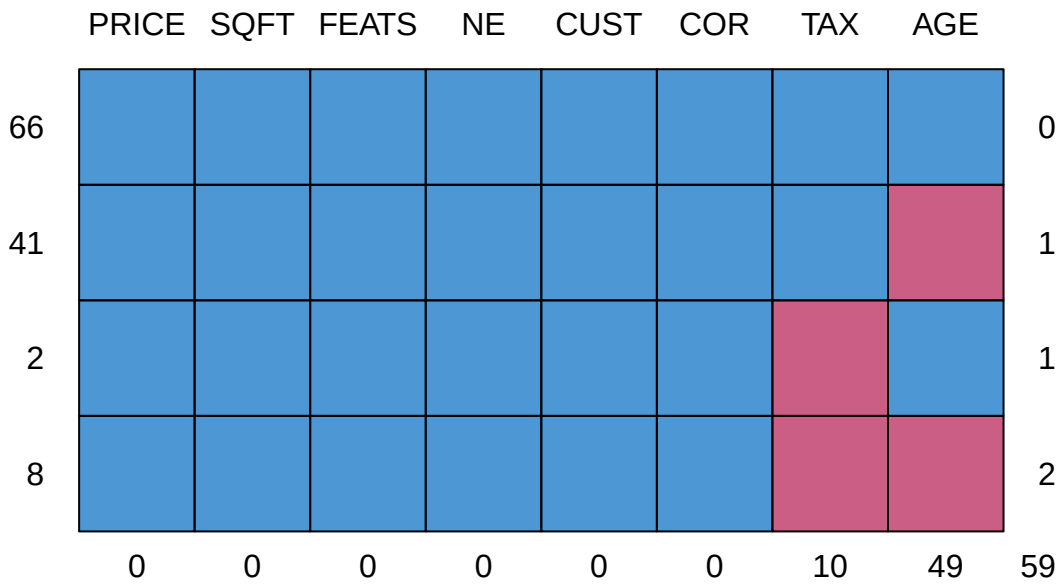
```
## Warning: 59 parsing failures.
## row col   expected actual          file
##   2 AGE an integer      * 'maisons.txt'
##   9 AGE an integer      * 'maisons.txt'
##   9 TAX an integer      * 'maisons.txt'
##  16 AGE an integer      * 'maisons.txt'
##  20 AGE an integer      * 'maisons.txt'
## ... ... ......... ...... ............
## See problems(...) for more details.
```

```
str(maisons)
```

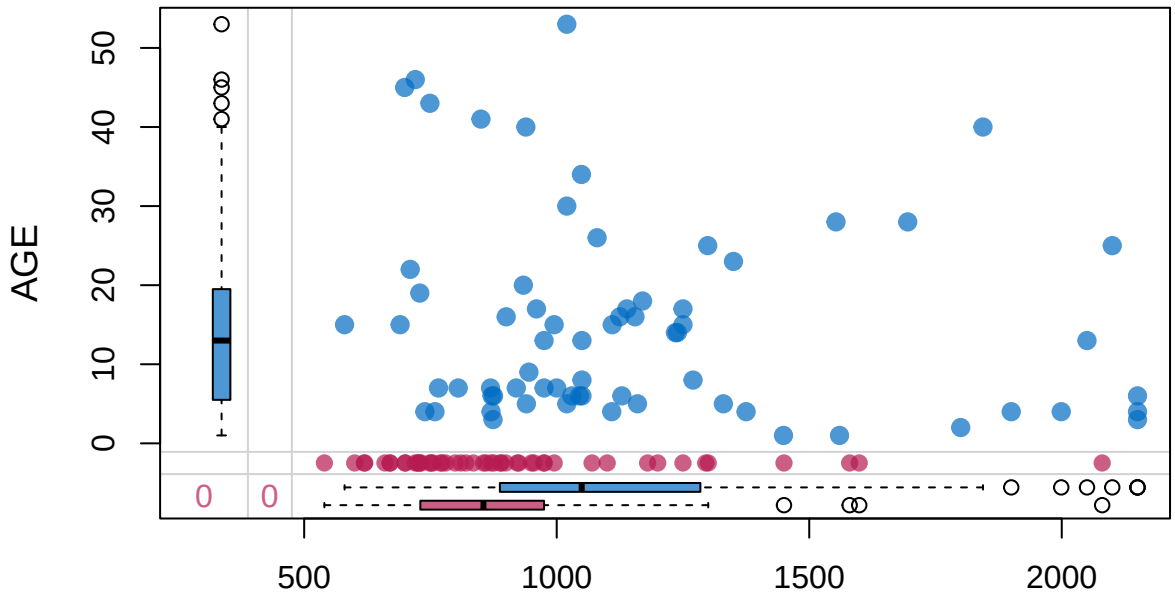
```
dim(maisons)
```

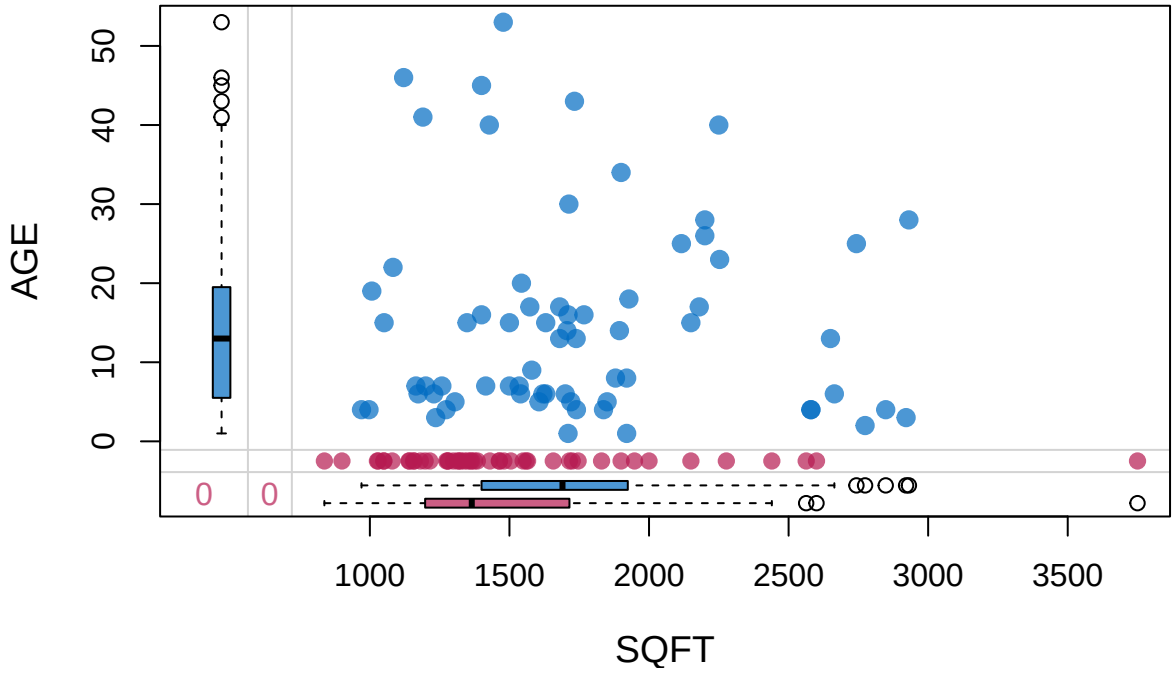
```
summary(maisons)
```

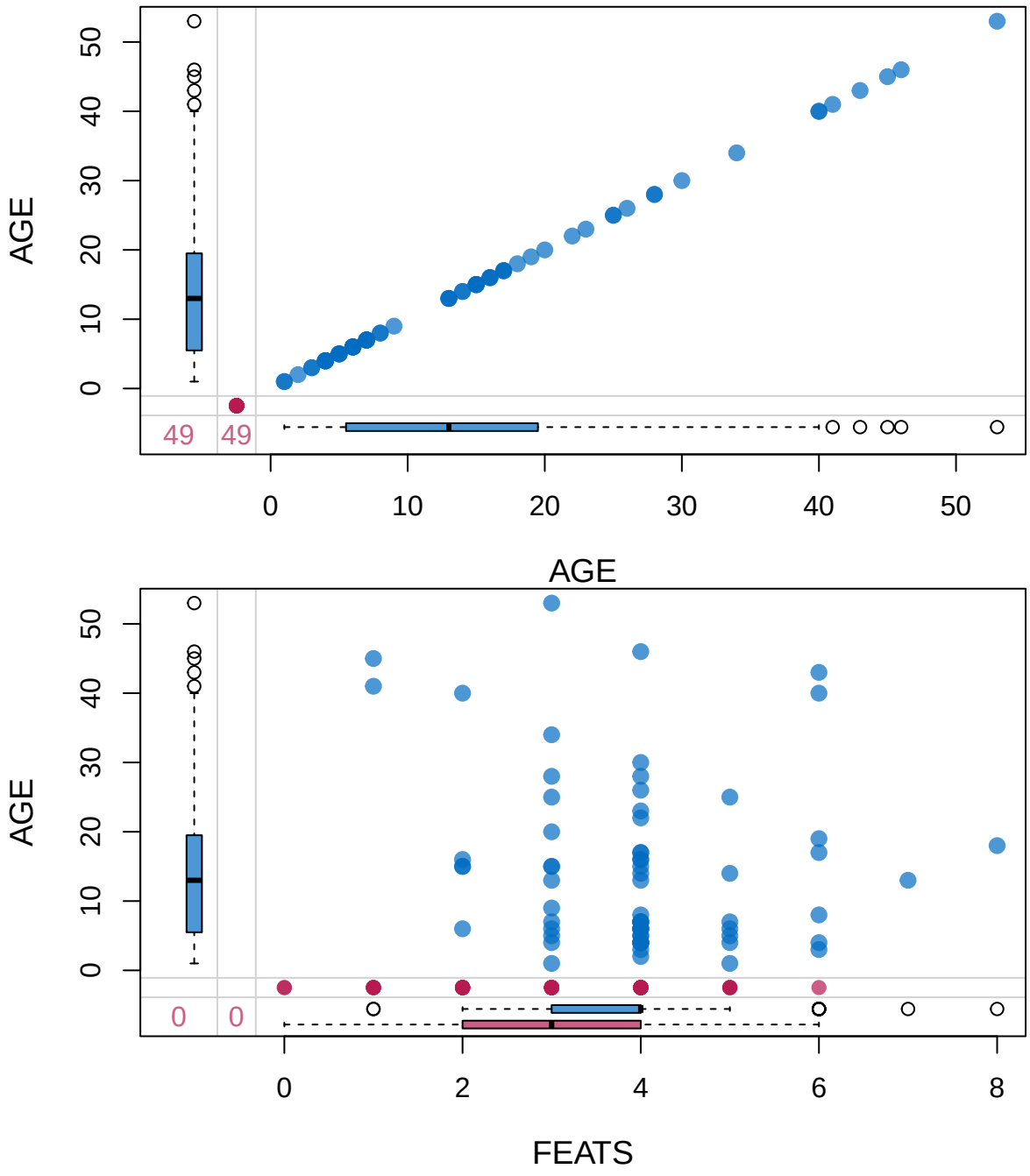
```
library(mice)
md.pattern(maisons)
```

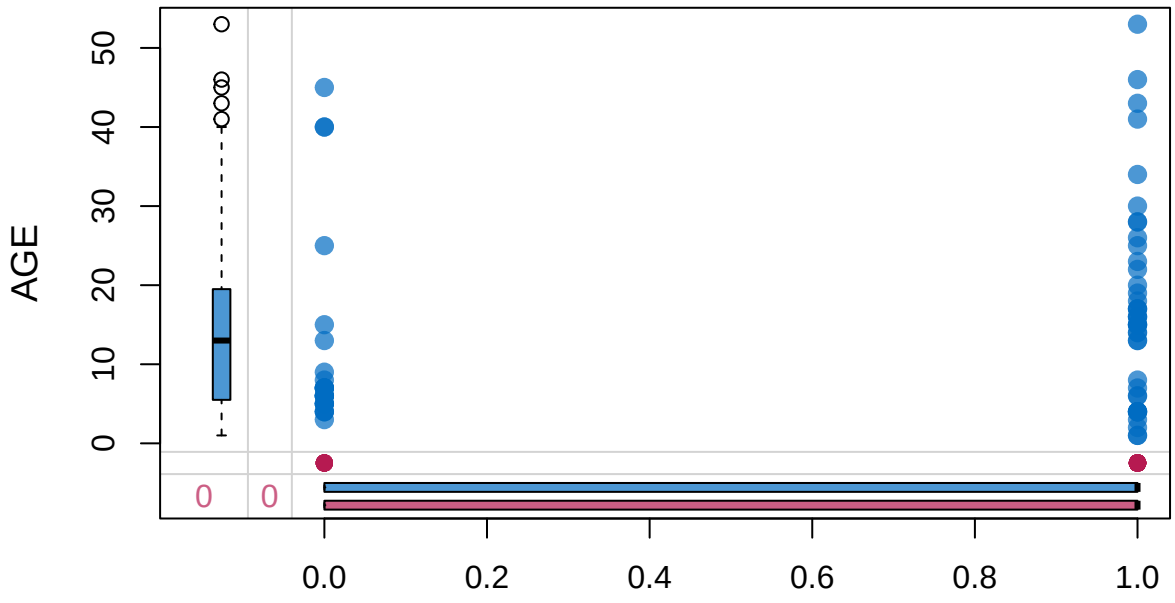


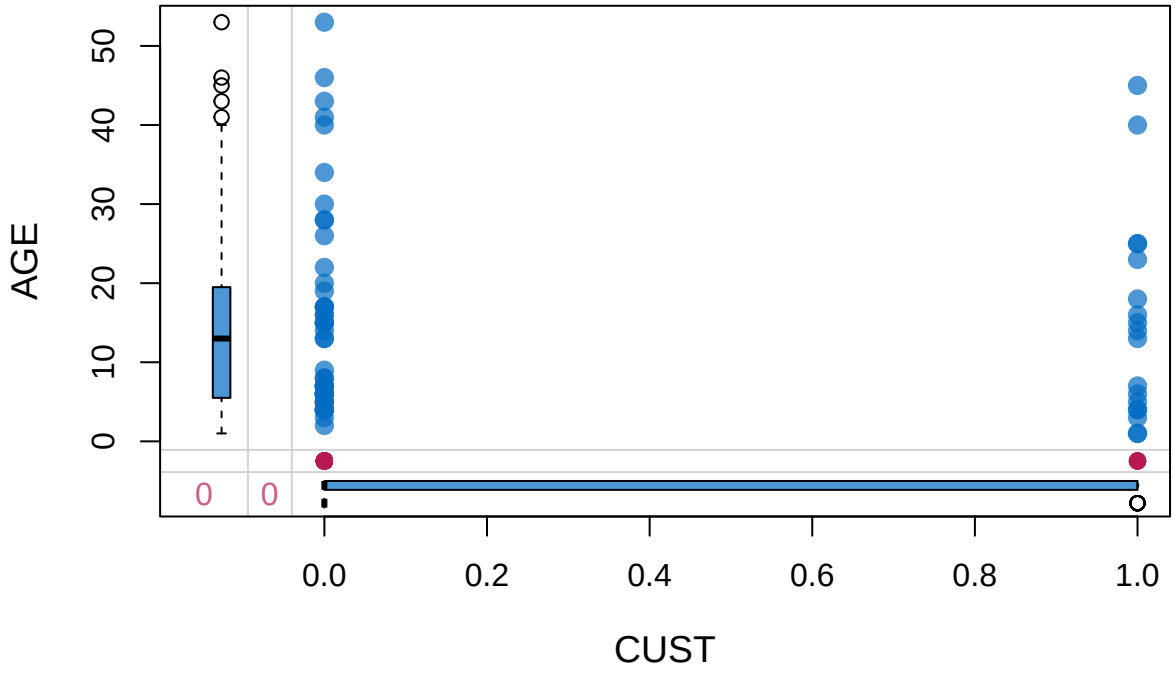

```
## MCAR, MAR ou MNAR
#install.packages("VIM")
library(VIM)

for (var in colnames(maisons)){
  marginplot(maisons[, c(var,"AGE")], col = mdc(c("obs", "mis")), cex = 1.2, cex.lab = 1.2,pch=19)
}
```

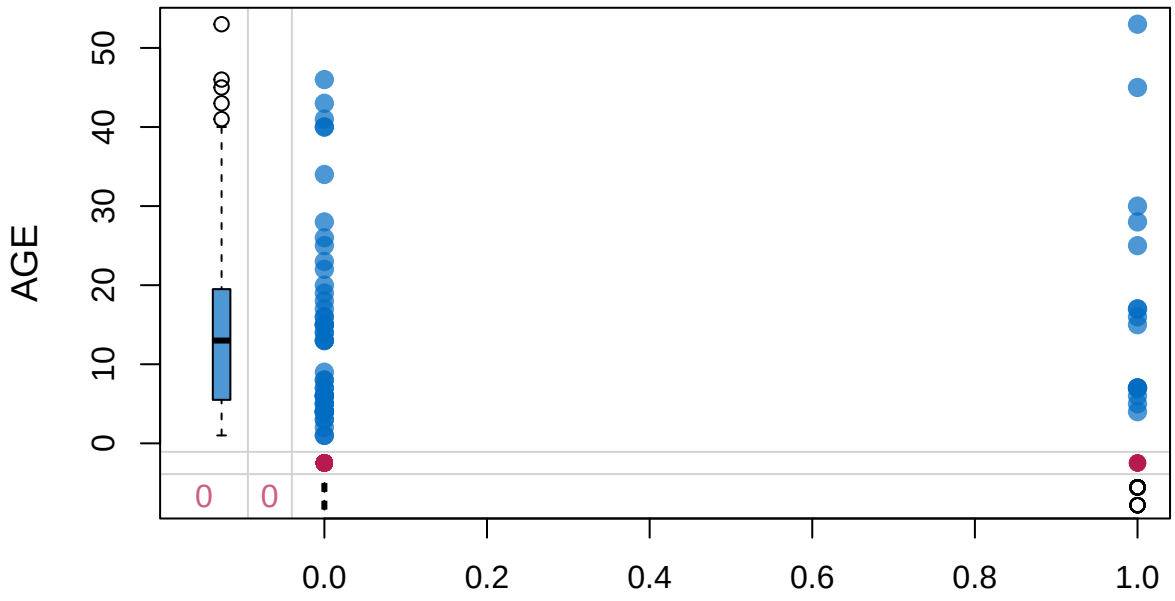

AGE
0
10
20
30
40
50
0
0
500
1000
1500
2000
PRICE

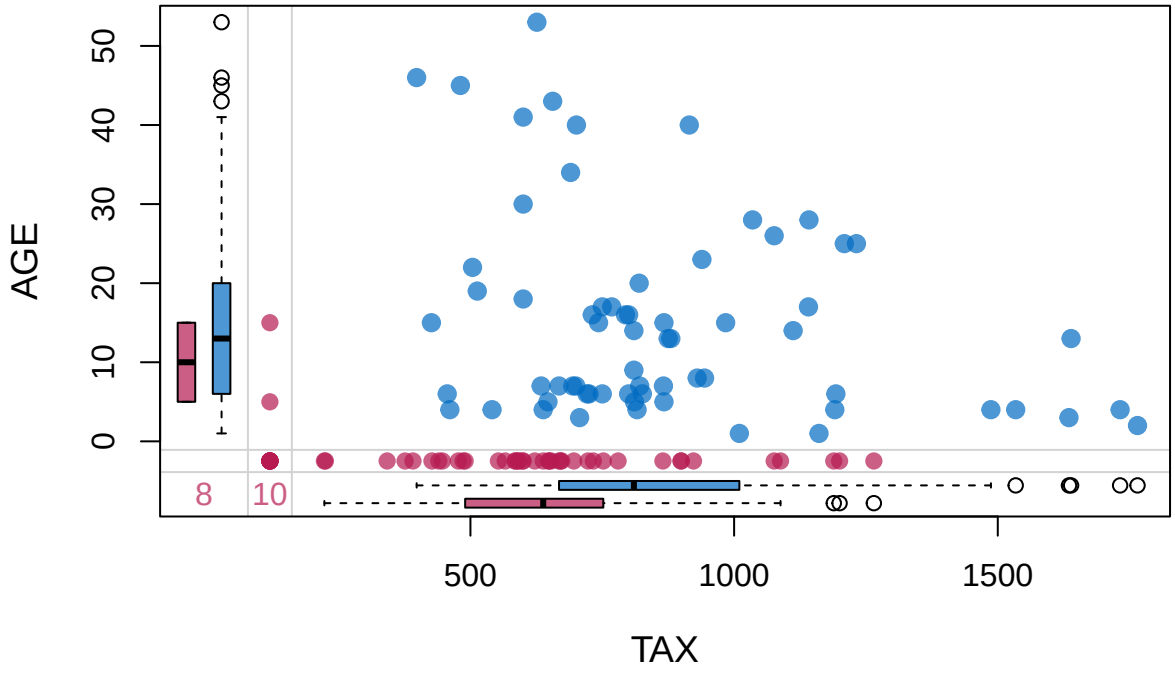

AGE
0
10
20
30
40
50
0
0
1000
1500
2000
2500
3000
3500
SQFT


AGE
0
10
20
30
40
50
49
49
0
10
20
30
40
50
AGE
AGE
0
10
20
30
40
50
0
0
0
2
4
6
8
FEATS


AGE
0
10
20
30
40
50
0
0
0.0
0.2
0.4
0.6
0.8
1.0
NE


AGE
0
10
20
30
40
50
0
0
0.0
0.2
0.4
0.6
0.8
1.0
CUST






```
t.test(maisons$PRICE[is.na(maisons$AGE)],maisons$PRICE[!is.na(maisons$AGE)])
```

```
t.test(maisons$FEATS[is.na(maisons$AGE)],maisons$FEATS[!is.na(maisons$AGE)])
```

```
t.test(maisons$SQFT[is.na(maisons$AGE)],maisons$SQFT[!is.na(maisons$AGE)])
```

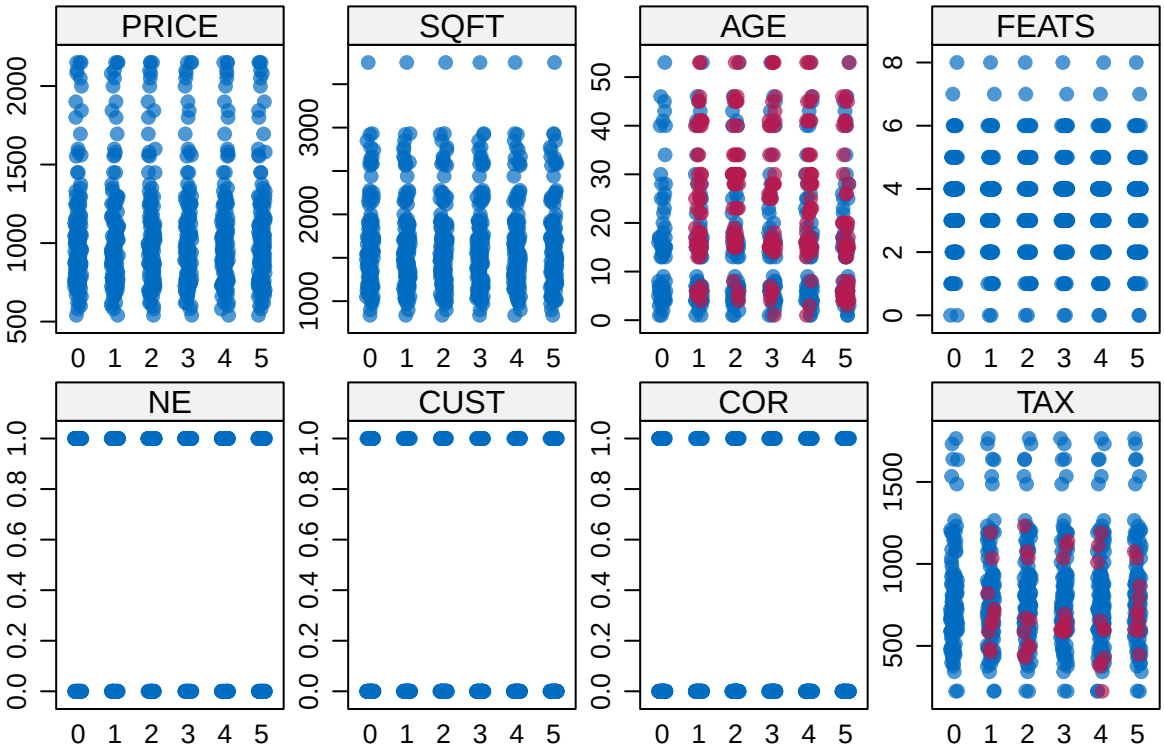
```
t.test(maisons$TAX[is.na(maisons$AGE)],maisons$TAX[!is.na(maisons$AGE)])
```

```
library(corrplot)
mcor <- cor(maisons[,1:8], use="pairwise.complete.obs")
print(round(mcor*100,2))
#corrplot(mcor, type="upper", order="hclust",tl.col="black", tl.srt = 45)
```

```
#MICE
# Charger le paquet pour l'imputation des données manquantes
library("mice")

dm <- mice(maisons, m=5, maxit=50, seed=123, print=FALSE)
```

```
d1 = complete(dm,1)
summary(d1)
```

```
library(lattice)
stripplot(dm, pch = 20, cex = 1.2)
```

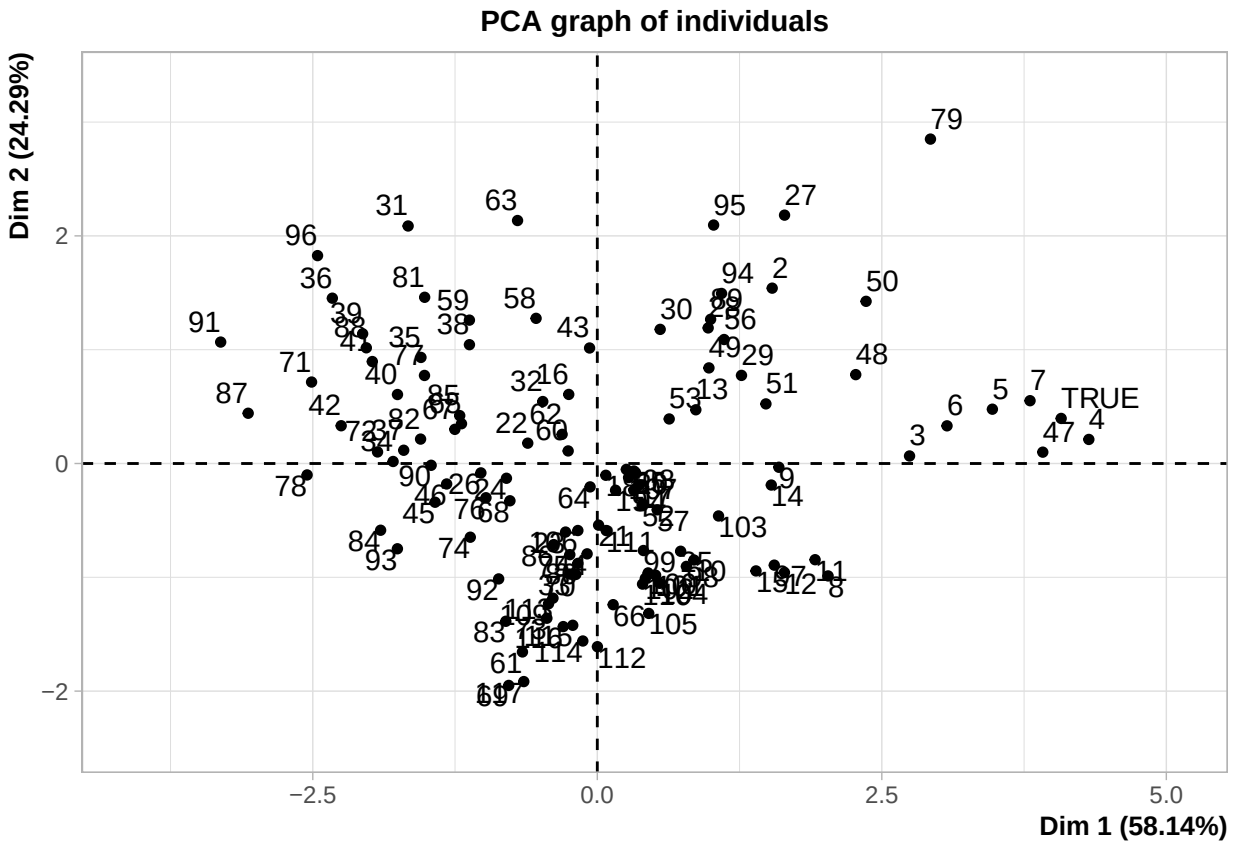


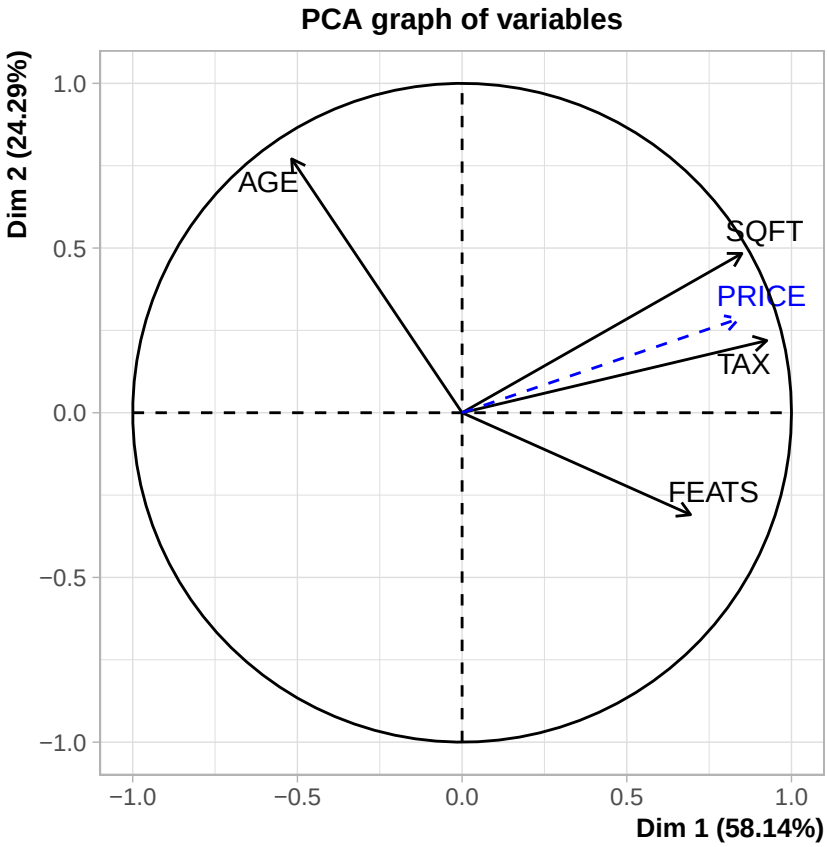

```
#Fusion des 5 jeux de données imputés
moy_complete = function (mice_ds){
  res = data.frame(row.names = T)
  for (col in colnames(mice_ds$'1')){
    for (i in 1:nrow(mice_ds$'1')) {
      res[i, col] = mean(c(mice_ds$'1'[i, col], mice_ds$'2'[i, col], mice_ds$'3'[i, col], mice_ds$'4'[i
    }
  }
  res[,"NE"] = as.logical(res[, "NE"])
  res[,"COR"] = as.logical(res[, "COR"])
  res[,"CUST"] = as.logical(res[, "CUST"])
  return (res)
}
```

```
dm_complete = complete(dm, "all")
dm_moy_complete = moy_complete(dm_complete)
```

```
#Etude de la multicolinéarité
library(FactoMineR)
```

```
acp = PCA(dm_moy_complete[,c(1,2,3,4,8)], scale=T, graph = T, quanti.sup = 1) #-7 : sans la variable PR
```






```
print(acp$eig)
```

```
fit <- lm(PRICE ~ ., dm_moy_complete)
print(fit)
```

```
library(car)
vif(fit)
```

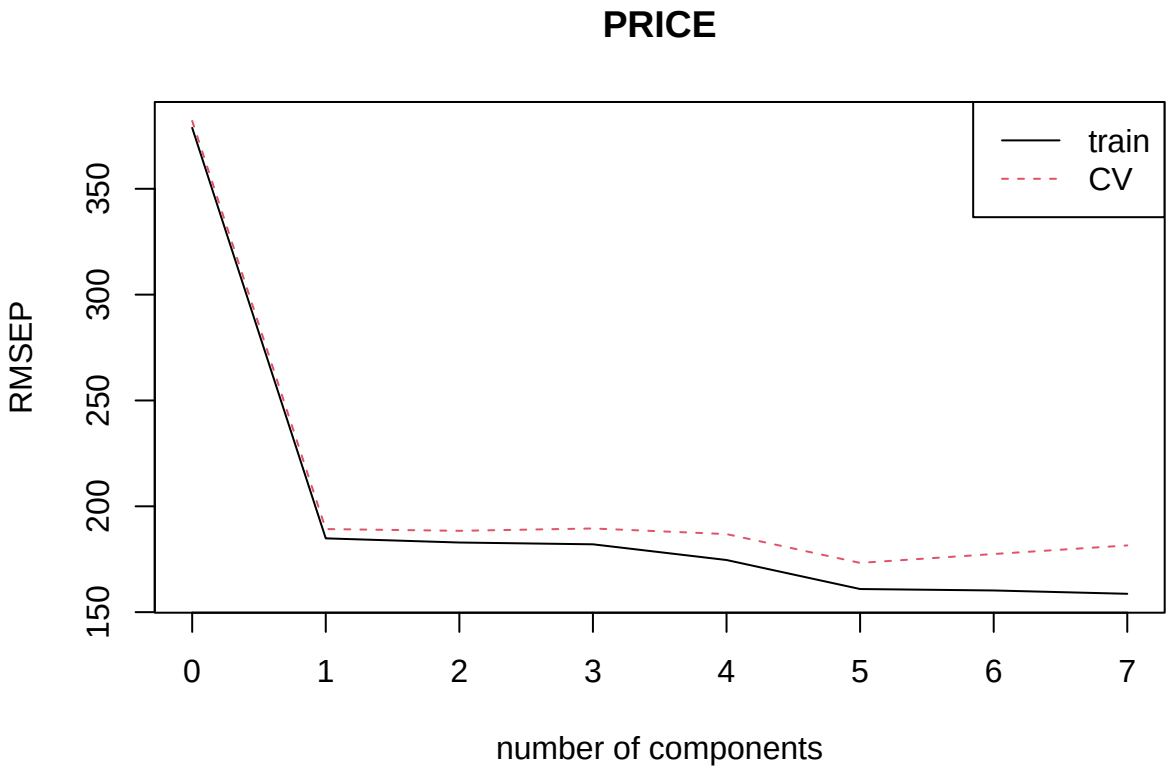
```
#Selection des variables (Ridge, Lasso, PCR, PLS, Elastic-net et stepAIC)

## stepAIC
library(MASS)

fit = lm(PRICE ~ ., dm_moy_complete)

aic<- stepAIC(fit, trace=0)
aic
```

```
rmsep=function(fit){
  h=lm.influence(fit)$h
  return(sqrt(mean((residuals(fit)/(1-h))^2)))
}

rmsep(aic)
```

PCR

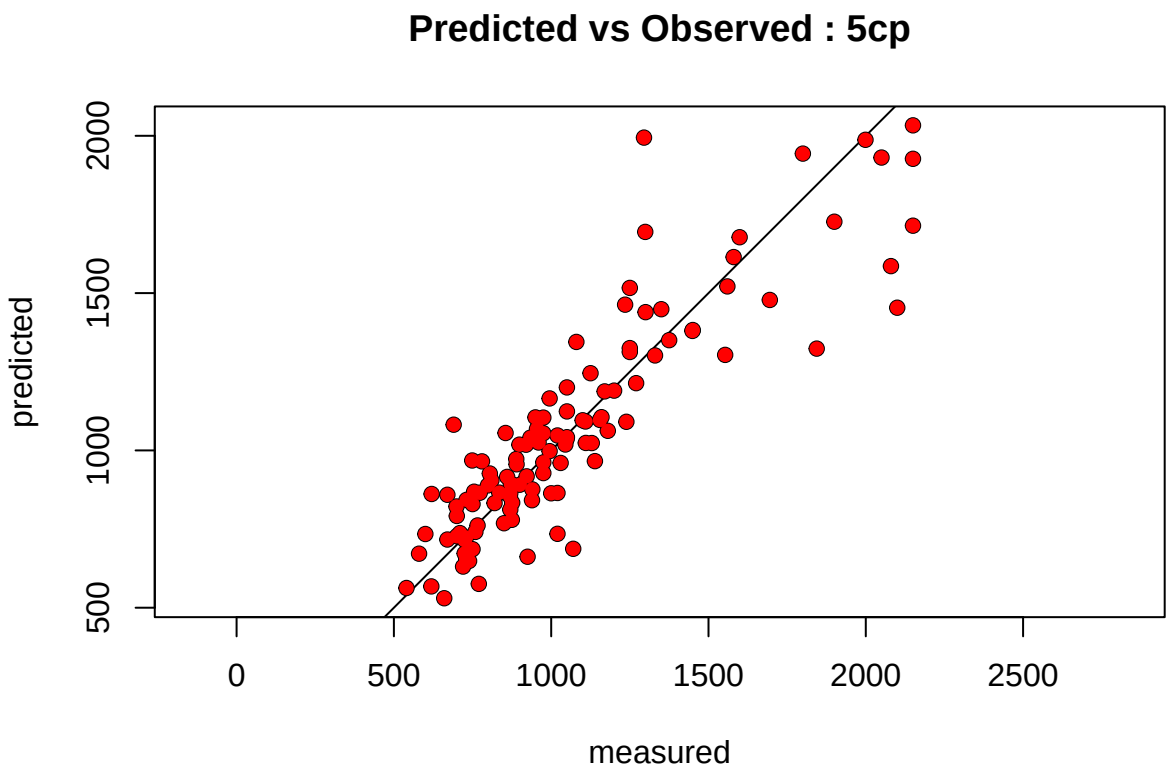
```
library(pls)

m_pcr=pcr(PRICE~., scale=TRUE, validation="LOO", jackknife = TRUE, data=dm_moy_complete)
```

```
plot(m_pcr,"validation", estimate = c("train", "CV"), legendpos = "topright")
```

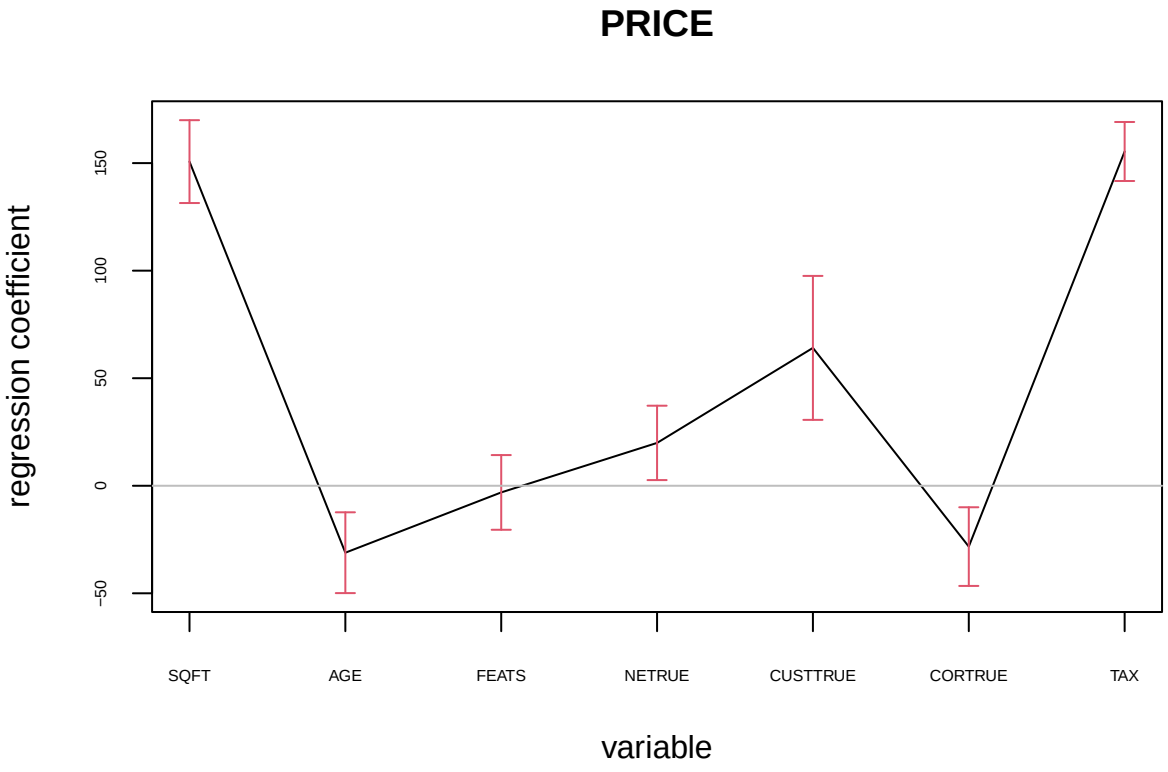



```
summary(m_pcr)
```

```
obsfit = predplot(m_pcr, ncomp=5, which = "validation", asp=1,line=TRUE, main="Predicted vs Observed : 5
points(obsfit, pch=16, col="red")
```




```
coefplot(m_pcr, ncomp=5, se.whiskers = TRUE, labels = prednames(m_pcr), cex.axis = 0.5)
```




```
jack.test(m_pcr, ncomp=5)
```

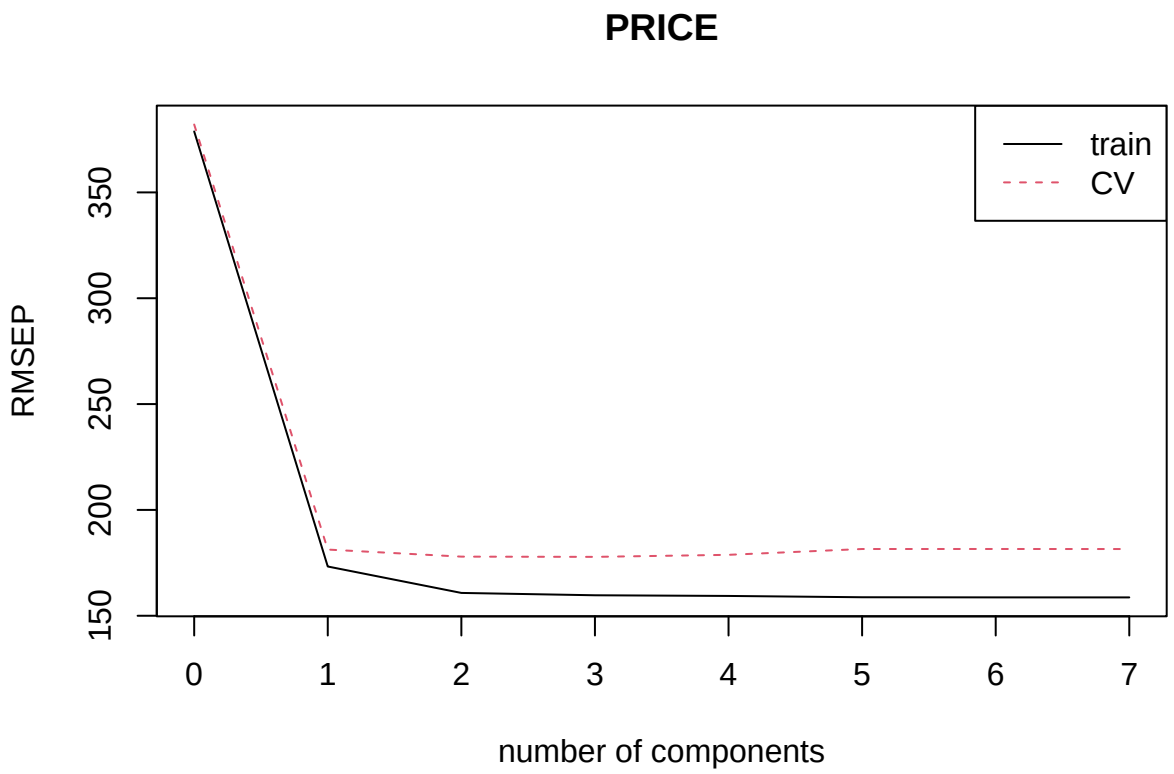
```
coef(m_pcr, ncomp=5, intercept=TRUE)
```

Les variables qui ont le plus d'importance dans le modèle obtenu sont les variables "SQFT", "TAX" et "CUST". Nous pouvons noter que ces variables sont égalmenet dans le modèle issu de stepAIC.

```
sds = apply(dm_moy_complete,2, "sd" ) # calcul des ecart-types de chaque variable
coef(m_pcr, ncomp=5, intercept=TRUE)[2:8]/sds[2:8]
```

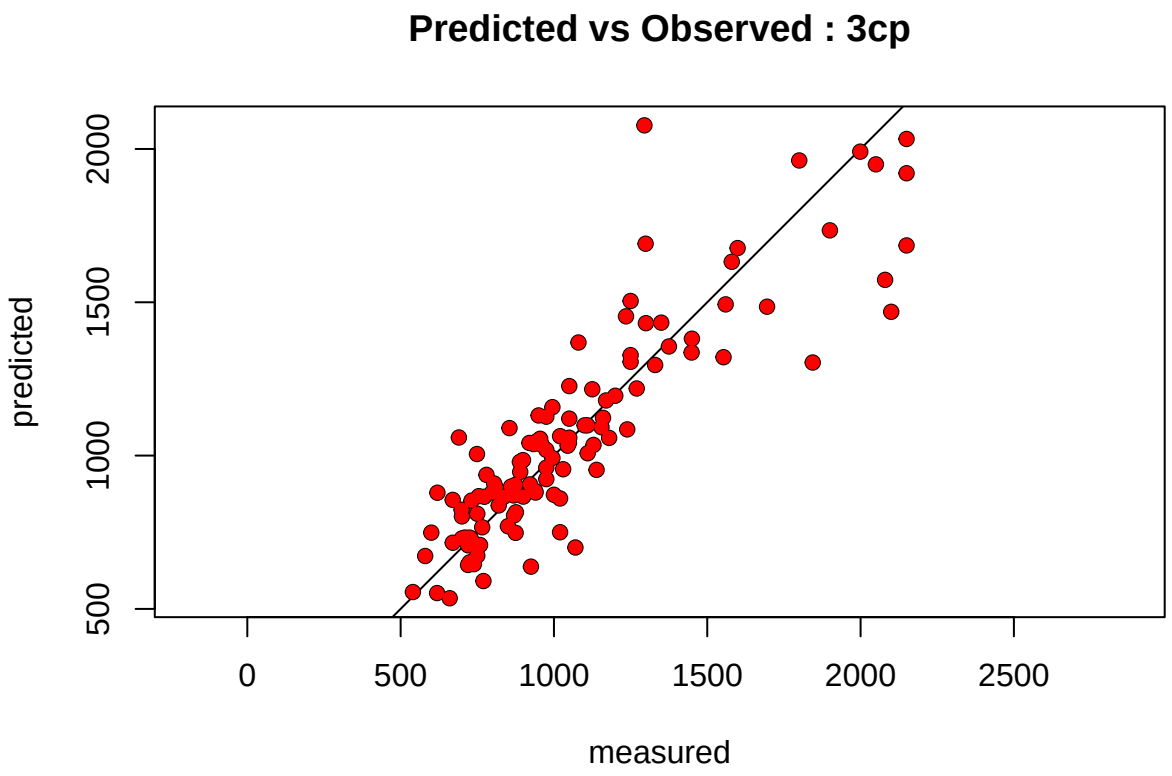
```
##PLS
m_pls=plsr(PRICE~., scale=TRUE, validation="LOO", jackknife = TRUE, data=dm_moy_complete)
```

```
plot(m_pls,"validation", estimate = c("train", "CV"), legendpos = "topright")
```

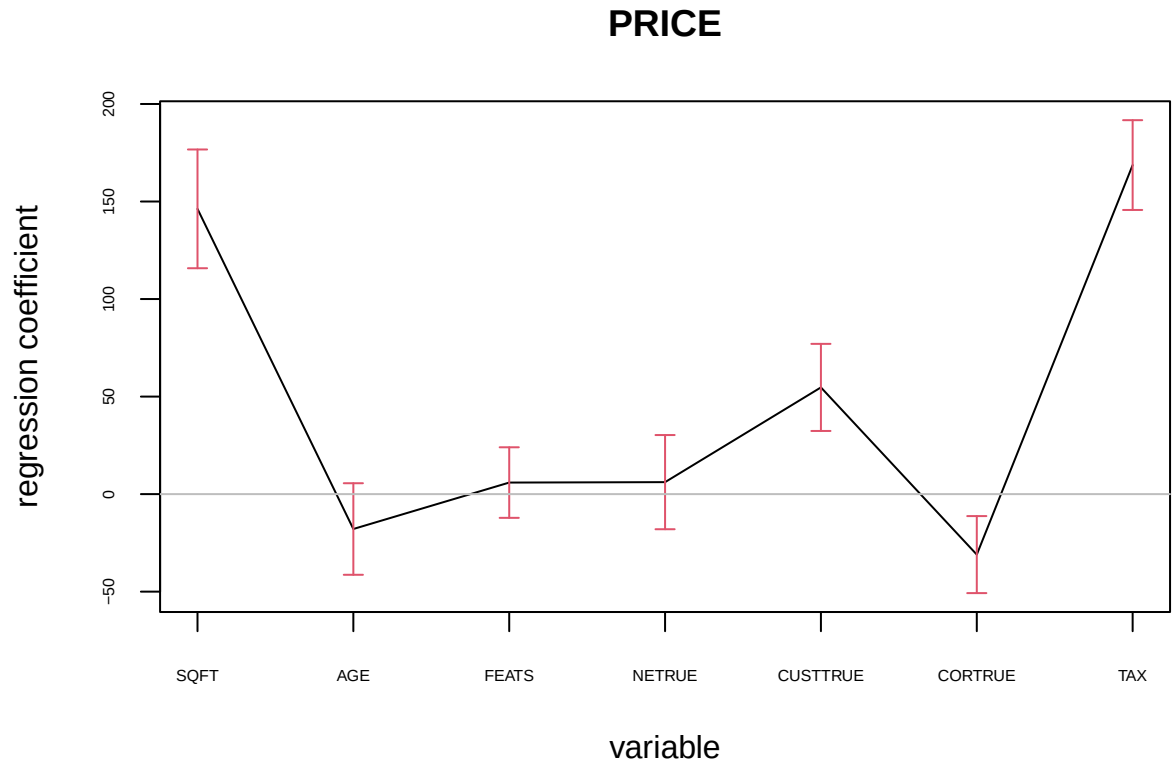



```
summary(m_pls)
```

```
obsfit = predplot(m_pls, ncomp=3, which = "validation", asp=1,line=TRUE, main="Predicted vs Observed : 3
points(obsfit, pch=16, col="red")
```




```
coefplot(m_pls, ncomp=3, se.whiskers = TRUE, labels = prednames(m_pcr), cex.axis = 0.5)
```



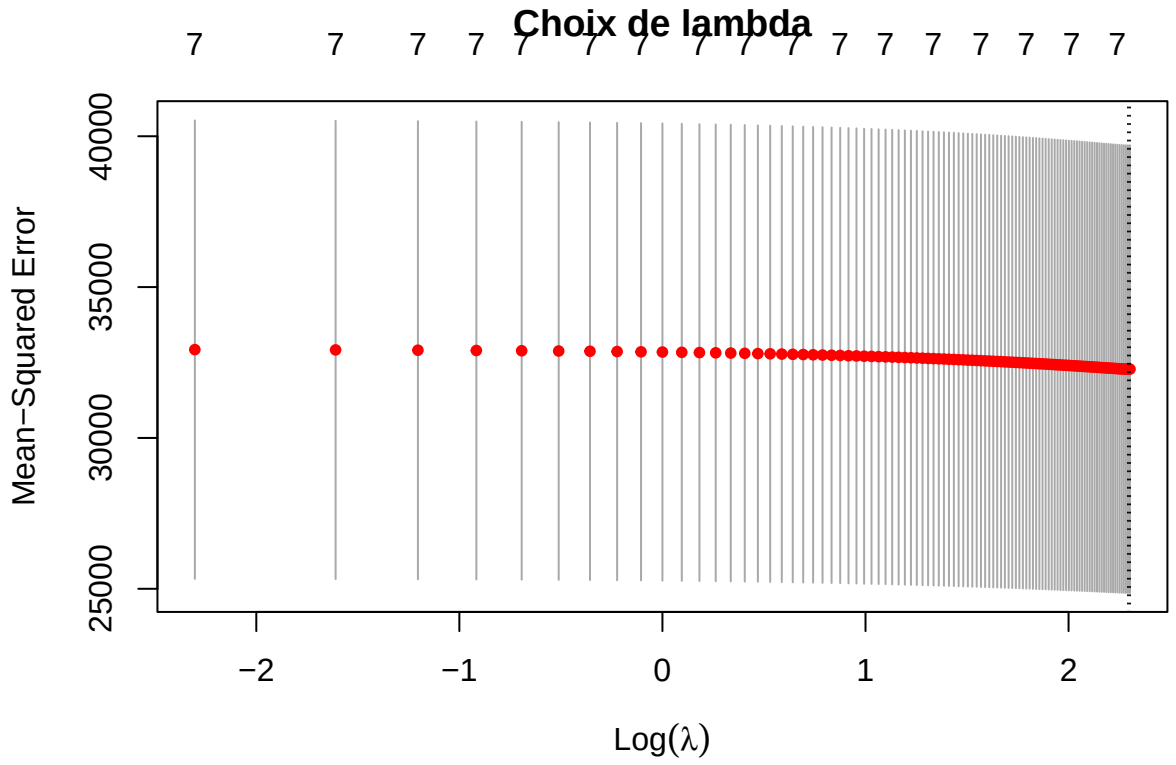

```
jack.test(m_pls, ncomp=3)
```

```
coef(m_pls, ncomp=3, intercept=TRUE)
```

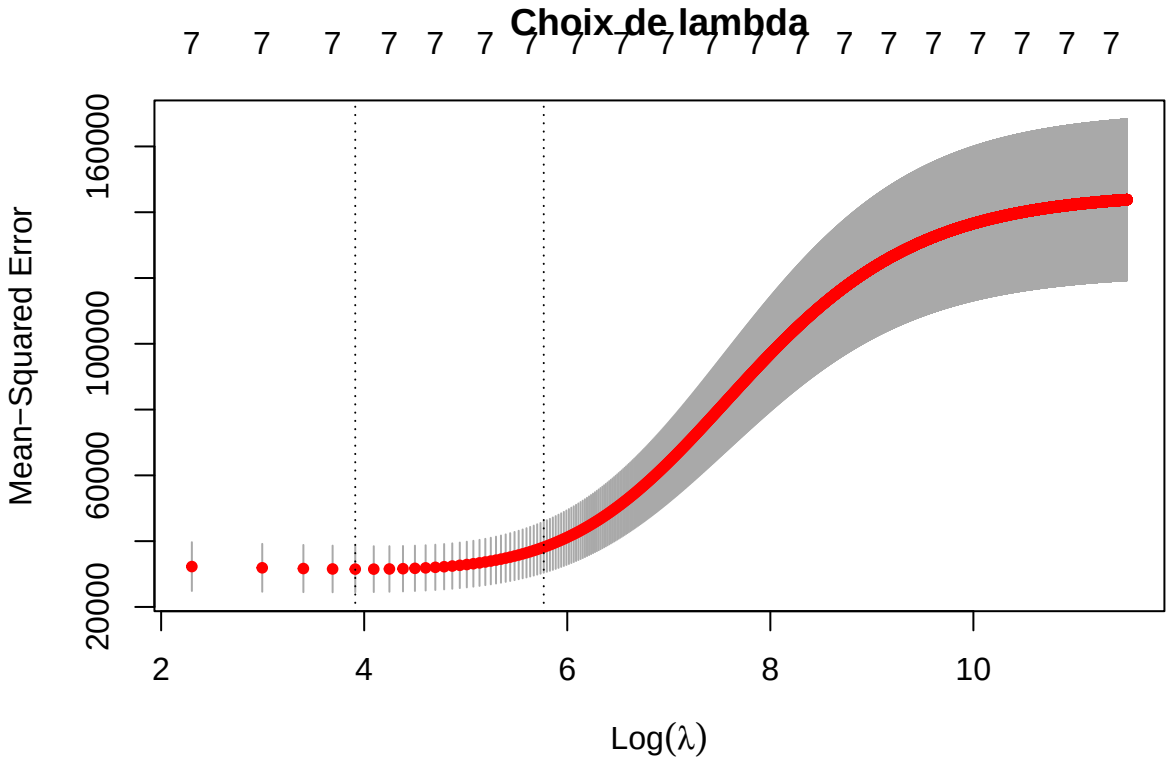
```r
sds = apply(dm_moy_complete,2, "sd" ) # calcul des ecart-types de chaque variable
coef(m_pls, ncomp=3, intercept=TRUE)[2:8]/sds[2:8]
```

```r
## Ridge
X = model.matrix(PRICE~., dm_moy_complete)[,-1]
Y =dm_moy_complete$PRICE
```

```r
library(glmnet)
#choisisons le lambda qui minimise le RMSEP (ou equivalent, la cross-validation = MSEP):
cv_fit <- cv.glmnet(X,Y, alpha = 0, lambda = seq(0,10, 0.1), grouped = FALSE, nfolds =nrow(dm_moy_compl
plot(cv_fit, main = "Choix de lambda")
```




```r
cv_fit <- cv.glmnet(X,Y, alpha = 0, lambda = seq(0,100000, 10), grouped = FALSE, nfolds =nrow(dm_moy_co
plot(cv_fit, main = "Choix de lambda")
```




```
lambda_optimal = cv_fit$lambda.min
print(lambda_optimal)
```

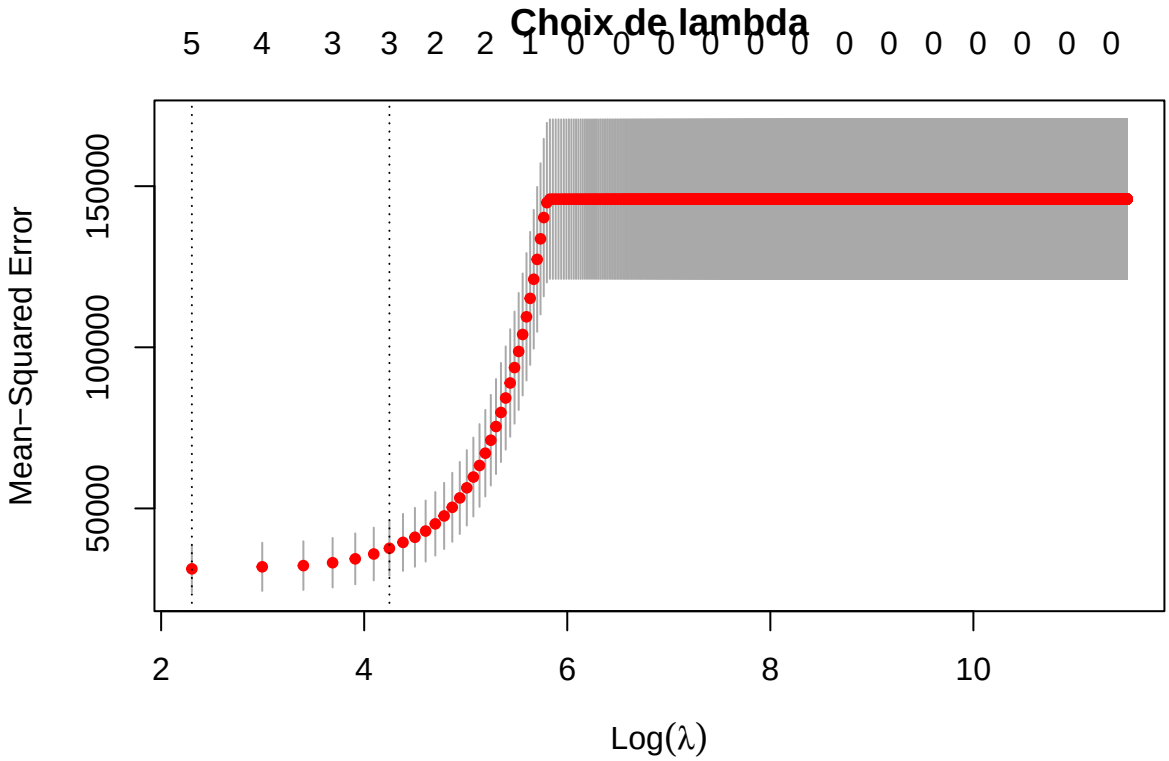
```
print(min(cv_fit$cvm))
```

```
rmsep_ridge =sqrt(min(cv_fit$cvm))
print(rmsep_ridge)
```

```
m_ridge <- glmnet(X,Y, alpha = 0, lambda = lambda_optimal)
```

```
### Coefficients sur les données d'origine
coef(m_ridge) # pour voir les coefficients
```

```
## Lasso
cv_fit <- cv.glmnet(X,Y, alpha = 1, lambda = seq(0,100000, 10), grouped = FALSE, nfolds =nrow(dm_moy_co
plot(cv_fit, main = "Choix de lambda")
```




```
lambda_optimal = cv_fit$lambda.min
print(lambda_optimal)
```

```
print(min(cv_fit$cvm))
```

```
### Coefficients sur les données normalisés
sds = apply(dm_moy_complete,2, "sd" )
coef(m_ridge)[2:8]*sds[2:8]
```

```
rmsep_lasso =sqrt(min(cv_fit$cvm))
print(rmsep_lasso)
```

```
m_lasso <- glmnet(X,Y, alpha = 1, lambda = lambda_optimal)
```

```
### Coefficients sur les données normalisés
coef(m_lasso)[2:8]*sds[2:8]
```

```
### Coefficients sur les données d'origine
coef(m_lasso)
```

```
## Elastic-net
elastic_net = function(X, Y, alpha_start, alpha_end, step){
  d = data.frame(matrix(ncol = 3))
  colnames(d) = c("alpha", "rmsep", "lambda")
  alpha = alpha_start
  i = 1
  while(alpha < alpha_end){
    cv_fit <- cv.glmnet(X,Y, alpha = alpha, grouped = FALSE, nfolds =nrow(X))
    lambda = cv_fit$lambda.min
    rmsep =sqrt(min(cv_fit$cvm))
    d[i,] = c(alpha, rmsep, lambda)
    alpha <- alpha + step
```

```
    i = i + 1
  }
  return (d)
}


en1 = elastic_net(X, Y, 0.1, 0.9, 0.1)
ind_min = which.min(en1$rmsep)

en1[(ind_min-1):(ind_min+1),]
```

```
en2 = elastic_net(X, Y, 0.3, 0.5, 0.01)
ind_min = which.min(en2$rmsep)

en2[(ind_min-1):(ind_min+1),]
```

```
en3 = elastic_net(X, Y, 0.41, 0.43, 0.001)
ind_min = which.min(en3$rmsep)

en3[(ind_min-1):(ind_min+1),]
```

```
m_elastic <- glmnet(X,Y, alpha = 0.422, lambda = 25.2738)
coef(m_elastic)
```